A comparative study of optimal stratification in business and agricultural surveys

---

A thesis

submitted in partial fulfilment

of the requirements for

the Degree of

Master of Science in Statistics

at the

University of Canterbury

by

Michael Clifford Hayward

University of Canterbury

2010

For Sylvia

Acknowledgements

I would first like to thank my primary supervisor Associate Professor Jennifer Brown, Head of the Department of Mathematics and Statistics at the University of Canterbury, for her support, understanding, and guidance during the process of conducting this research. I am also indebt to Richard Penny, Senior Methodologist at Statistics New Zealand, for suggesting the area of research, and for his insight into some of the practical issues faced in survey statistics.

I am grateful for the financial support received through a University of Canterbury Masters Scholarship and through a New Zealand Institute of Mathematics and its Applications (NZIMA) Centre of Research Excellence Postgraduate Scholarship. I also appreciated the support from the University of Canterbury Department of Mathematics and Statistics and the University of Otago Wellington School of Medicine and Health Sciences to present some of this work at a New Zealand Statistics Association (NZSA) conference and a Statistics Society of Australia Inc (SSAI) conference.

Finally I would like to thank my family, friends, and colleagues, for their support and understanding during the process of completing this work. In particular I have dedicated this thesis to my mother Sylvia Hayward for her

support and encouragement, and I hope that this will provide some inspiration in her own pursuit of higher education in years to come.

Abstract

This thesis is a comparative study of optimal design-based univariate stratification as applied to highly skewed populations such as those observed in business and agricultural surveys. Optimal stratification is a widely used method for reducing the variance or cost of estimates, and this work considers various optimal stratification algorithms, and in particular optimal boundary algorithms, to support this objective.

We first provide a background to the theory of stratification and stratified random sampling, and extend this through the derivation of optimal allocation strategies. We then examine the effect of allocation strategies on the variance and design effect of estimators, and in particular find several issues in applying optimal or Neyman allocation when there is little correlation between the survey population and auxiliary information.

We present a derivation of the intractable equations for the construction of optimal stratum boundaries, based on the work of Dalenius (1950), and derive the cumulative square root of frequency approximation of Dalenius & Hodges (1957). We then note a number of issues within the implementation of the cumulative square root of frequency rule surrounding the construction of initial intervals, and find that the placement of boundaries and the variance

of estimates can be affected by the number of initial intervals. This then leads us to propose two new extensions to the cumulative square root of frequency algorithm, using linear and spline interpolation, and we find that these result in some improvements in the results for this algorithm.

We also present a complete derivation of the Ekman algorithm, and consider the extended approach of Hedlin (2000). We derive several new results relating to the Ekman algorithm, and propose a new kernel density based algorithm. We find all three Ekman based algorithms produce similar results for larger populations, and provide some recommendations on the use of these algorithms depending on the size of the population.

We look at the derivation and implementation of the Lavallée-Hidiroglou algorithm, and find that it is often slow to converge or does not converge for Neyman allocation. We therefore adopt a random search model of Kozak (2004), and note that the Lavallée-Hidiroglou algorithm generally produces superior results across all populations used in this thesis.

We briefly investigate the optimal number of strata by examining the work of Cochran (1977) and Kozak (2006), and find that there is a diminishing marginal effect from increasing the number of strata and possibly some benefit from constructing more than six strata. However we also acknowledge that the cost of constructing such strata may offset any potential gain in precision from constructing more than five or six strata.

Finally we consider the how many of these problems can be developed further, and ultimately find that such problems for deciding the number of strata, construction of stratum boundaries, and the allocation of sample units among the strata may require an approach that takes account of the

relationship between the auxiliary variable and the survey information. We therefore suggest investigating these algorithms further within the context of a model-assisted environment in order to help account for the relationship between the auxiliary information and survey population.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Design-based optimal stratification is a widely used sample survey method to minimise the variance of population estimates, minimise the cost of sampling, and reduce the response burden on survey participants. It is a particularly important technique for highly skewed populations, such as those present in business and agricultural surveys, in order to derive efficient and effective sample estimates.

Optimal stratification is usually separated into three optimisation problems: the number of strata to construct, the placement of stratum boundaries, and the number of observations to be selected from each stratum. Notable contributions to these problems include Cochran (1977) on the number of strata, Dalenius (1950), Dalenius & Hodges (1957), Ekman (1969), and Lavallée & Hidiroglou (1988) on the placement of stratum boundaries, and Neyman (1934) on the number of observations to be selected from each stra-

tum.

Dalenius (1950) first provided equations for the determination of stratum boundaries that minimise the variance of population estimates under optimal allocation; however he also acknowledged that these are troublesome to solve. One proposed solution to these equations for the optimal stratum boundaries is to take equal intervals of the cumulative square root of frequency scale of the stratification variable (Dalenius & Hodges 1957). Since then there has been a considerable number of approximations proposed for the construction of stratum boundaries that may provide better or faster solutions to many of the optimisation problems, with most making important assumptions such as the uniform distribution of values within strata. Unfortunately some of these assumptions may limit the application of these algorithms to highly skewed populations.

Optimal stratification also requires good auxiliary (or supplementary) information to assist in solving many of these optimisation problems. There has been a proliferation of such information since much of the above foundational work on stratification, considerable increases in computational capability, and interest in the application of algorithms to the stratification of highly skewed populations such as those encountered in business and agricultural surveys (Rivest 2002).

This thesis will investigate design- (or randomisation-) based univariate stratification, and in particular the construction and placement of optimal boundaries. It is primarily a comparative analysis of optimal stratification algorithms as applied to the highly skewed populations that are often encountered in business and agricultural surveys. The work will also attempt to

implement the various optimal stratification algorithms using the R programming language, and in doing so test some of the assumptions that underpin the various approaches.

This first chapter gives a brief overview of stratification, and some of the ideas behind optimal stratification. It then outlines some of the problems encountered with highly skewed populations in business and agricultural surveys, and gives a brief overview of the approach and material covered in the remainder of the thesis.

1.2 Stratification

Stratification in statistics is a process by which a statistical population is divided into mutually exclusive subpopulations called strata. This can be expressed using set notation for population $U = \{1, \ldots, N\}$ divided into strata U_i as follows (Horgan 2006):

$$U = \bigcup_{i=1}^{L} U_i, \quad U_i \cap U_j = \emptyset, \quad i \neq j \in \{1, \ldots, L\}$$

Each individual or unit within the population is assigned to one of the strata, and no unit resides outside of one of these strata ($U = \bigcup_{i=1}^{L} U_i$). Likewise no unit is assigned to more than one strata ($U_i \cap U_j = \emptyset$).

This has convenient parallels to similar concepts of strata in other disciplines. For example sociology and other social sciences arrange individuals into social strata using demographic and socio-economic factors to explore inequalities between groups. Geology classifies layers of rock or soil into strata

for analysis, and biology can consider strata in the context of layers of tissue. Many such examples can also form the basis of stratification in statistics.

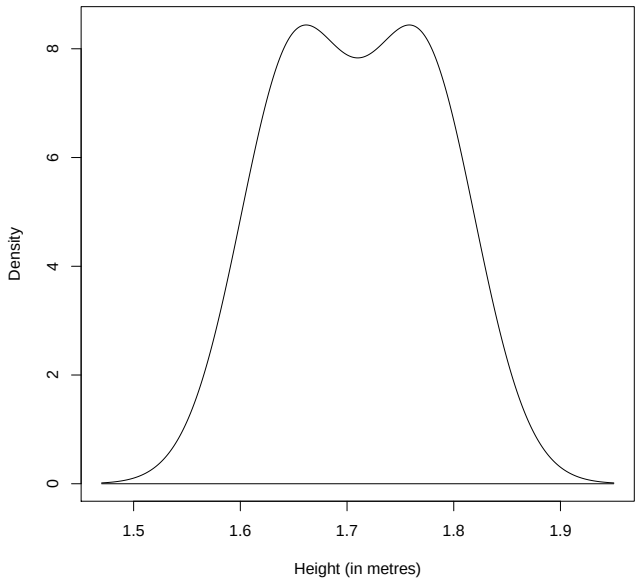

Figure 1.1: Distribution of height for 19 - 45 year old individuals in New Zealand

We can construct a simple example of stratification using the distribution of height. Figure 1.1 shows an overall bi-modal distribution of height for the New Zealand adult population, based on the estimates from Wilson, Russell & Wilson (1993). This can then be divided into two strata, males and females, to both better represent the overall population and derive meaningful information from each stratum.

The two strata in figure 1.2, labelled "Males" and "Females", comprise the entire population U. All of the males are included in the "Males" stratum and all of the females in the "Females" stratum. A sample can then be taken from each stratum, with a sample size that is proportional to the representation of males and females in the population, to help ensure an overall representative

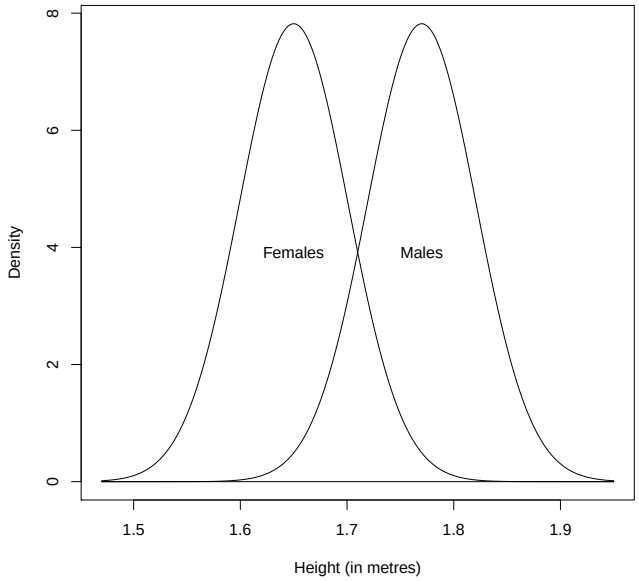

Figure 1.2: Distribution of height for 19 - 45 year old males and females in New Zealand

sample.

Populations are often divided into strata, and some of the main reasons for this include:

- Obtaining information about sub-groups, or strata. Stratification can be a convenient way to obtain information about subgroups as well as the overall population of interest. For example we may be interested in health and wellbeing of specific social-demographic groups within a population, as well as the overall population.

- Ensuring representation of subgroups within a sample. Stratification can help ensure that there is a similar representation of groups in a population within the sample. For example this could ensure an appropriate representation of males and females in a stratified design by

sex (as demonstrated in the example above), and hence avoid a really bad sample that includes a large number of males or females.

- Exploiting some administrative convenience. There may exist some administrative advantage to stratification, such as proximity to branches or interviewers that result in benefits from a stratification by the likes of geographic area to minimise the overall cost of conducting a survey or census.

- Improving the accuracy of the overall estimates. Stratification can be used to improve the overall results of a survey by constructing homogenous sub-populations to minimise variance within groups and hence improve overall population estimates.

Implicit in each of the above is some form of auxiliary (or supplementary) information and classification schema in order to stratify the population. For example strata may be constructed using ethnicity in social sciences to obtain information and analyse differences in health outcomes or educational attainment. Similarly the auxiliary information could be geographic location, using provincial or city council as possible classifications.

Optimal stratification concerns the fourth of the above points: to minimise the variance or cost of population estimates, usually to find the estimated population mean or total under a fixed sample size. The next section goes through some of the ideas and issues in optimal stratification, and sets the scene for the remainder of the work in this thesis.

1.3 Optimal Stratification

Optimal stratification is a form of stratification designed to improve the precision of estimates, reduce cost, or minimise the response burden. Within this there are three main issues or questions that an optimal stratification design needs to address:

- The number of strata that should be used
- The construction and placement of stratum boundaries
- The allocation of sample units among the strata

Sometimes a fourth issue is noted regarding the total number of units to sample; however we will see in chapter 3 that this is linked to the sampling objective, such as minimising the variance of estimators for a given cost, and hence derived from the allocation of sample units among the strata.

Neyman (1934) calculated the equations for the allocation of sample units among the strata, for given information on the stratum population size and variance, in order to minimise the variance of sample estimates. This has since been further extended to incorporate and account for information on the cost of sampling from each stratum and is usually referred to as "optimal allocation" (Stuart 1954). We review some of the more significant results for optimal allocation in design-based optimal stratification in chapter 3.

Cochran (1977) has also investigated the reduction in variance from increasing the number of strata, finding that there is little to be gained from more than six strata unless the correlation between the auxiliary information

and the sample population is greater than 0.9. We likewise investigate and review the work on selecting the number of strata in chapter 9.

The equations for the construction of optimal (minimum variance or minimum cost) stratum boundaries using optimal (Neyman) allocation were first proposed by Dalenius (1950). However the equations for the optimal boundaries have considerable dependencies among the components, and were acknowledged computationally difficult to solve. Consequently there have been a number of approximate methods devised in order to arrive at a solution to these intractable equations, with one of the most significant of these approximations being the cumulative square root of frequency rule.

The cumulative square root of frequency is a widely used boundary algorithm first proposed by Dalenius & Hodges (1957). The algorithm assumes that the population within each stratum is approximately uniform, and then derives the optimal stratum boundaries by constructing equal intervals on the cumulative square root of frequency scale. This approximation therefore suggests that optimal boundaries are obtained by finding those boundary points that ensure the square root of the frequency is roughly the same between each stratum, and we investigate the cumulative square root of frequency rule further in chapter 5.

Two other significant approach's are the Ekman ($1959a$) algorithm and the Lavallée & Hidiroglou (1988) algorithm, both of which involve iterative procedures to find the optimal stratum boundaries. The Ekman algorithm uses an iterative procedure to equalise the product of the stratum weight and range to find the optimal stratum boundaries, and we go through the derivation and results for this algorithm in chapter 6. The Lavallée-Hidiroglou

iterative algorithm uses a series of differential equations to find the partial derivatives of the equation for the variance of the estimators, based on a procedure suggested by Sethi (1963), and we likewise look at this algorithm in chapter 7.

There have also been a number of other lesser known approaches, such as the geometric progression algorithm of Gunning & Horgan (2004). Another possibility is to construct strata by simply taking equal intervals over the range or the frequency of the population, and we go into the details for some of these in chapter 8. There are also other methods such as using boundaries calculated on standard distributions that may be similar to the population of interest (Sethi 1963); however some of these are beyond the scope of this thesis.

1.4 Business and Agricultural Surveys

Business and agricultural populations can create a number of challenging issues for survey sampling (Sigman & Monsour 1995):

- Populations tend to be highly skewed, with a small number of businesses accounting for a large proportion of the population total.
- Businesses are dynamic, with new businesses being created, existing businesses merging or closing, and businesses changing activities.
- There can be complex inter-business relationships, through varying partnerships, or through being owned by the same parent organisation.

Our principal concern for optimal stratification is the skewness of the population, and there are a number of other techniques to address the dynamic nature of businesses, and the interrelationships between businesses.

Optimal stratification results in large improvements to the variance (or cost) of estimators when a population is comprised of a number of similar groups. This is particularly the case for highly skewed populations, where there is a concentration of values around a particular point, often zero, and the presence of a number of very large values.

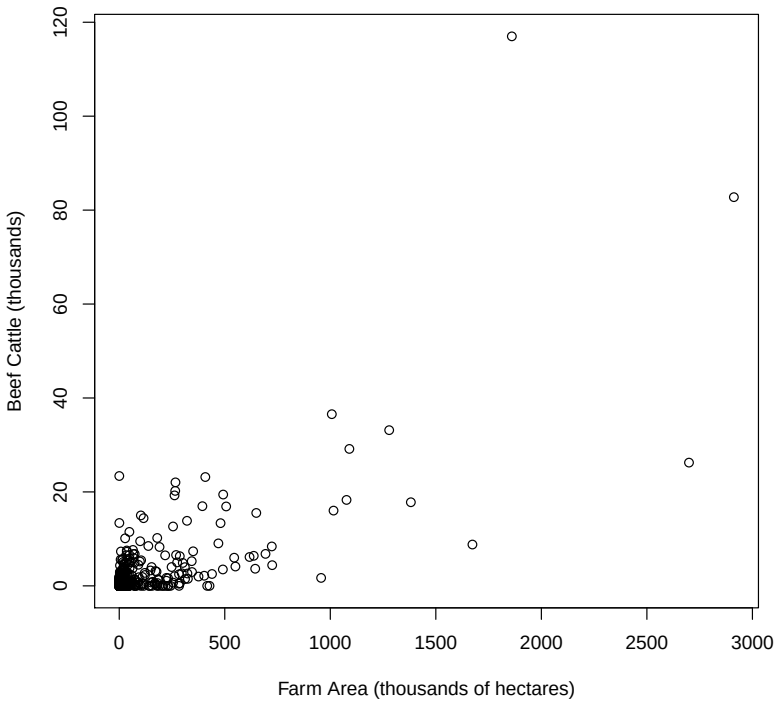

Figure 1.3: Scatter plot of Beef Cattle (thousands) and Farm Area (thousands of hectares) from the Australian Agricultural and Grazing Industries Survey (AAGIS)

Business and agricultural populations tend to be highly skewed, with a small number of very large businesses. An example of this is given by the

Australian Agricultural and Grazing Industries Survey (AAGIS) data in figure 1.3, where we see a small number of very large farms with a considerable number of beef cattle, and a large number of small farms with a very small number of beef cattle. In such a situation, the gains from optimal stratification could be considerable.

Stratification does require an appropriate auxiliary variable in order to divide the population into mutually exclusive strata and determine the number of units to sample from each stratum. However many such populations tend to have some easily obtainable auxiliary information, such as some measure of the size of the organisation. The availability of this auxiliary information has made optimal stratification one of the most widely used sample survey techniques for business populations.

1.5 Purpose and Approach

This thesis is an applied investigation and comparison of univariate design-based optimal stratification of highly skewed populations, such as those present in business and agricultural surveys. As such it will focus on the application of optimal stratification algorithms, and in particular optimal boundary algorithms.

The work is confined to a univariate approach, as much of the work on optimal stratification has concentrated on univariate stratification (Horgan 2006). Adopting such an approach means that we are able to focus of the relative merits of each algorithm, and provides a wider range of commonly used methods for optimal stratification (and in particular optimal boundary

algorithms).

We will also focus on design-based stratification, as most of the foundation work has likewise adopted a design-based approach. In particular the design-based work of Dalenius (1950) and Dalenius & Hodges (1957) is central to this thesis, and our interest in this work makes it appropriate to place similar limits of this thesis. It is a natural extension of much of this work to move into a model-assisted environment, but we are not able to give adequate consideration to this alongside the design-based approaches that are central to this thesis.

We will furthermore assume that there is a known auxiliary variable, and for the most part assume that it is the same as the sample population. Again this is a common assumption by other authors to ensure that sufficient limits are placed on the scope of the work under consideration, and to limit the effect of the correlation between the auxiliary information and the survey population on any analysis. There are a number of other works that focus on extending stratification by developing models of the relationship between auxiliary variables and the sample population (Sigman & Monsour 1995).

Optimal stratification can also exist as one of several stages of a complex survey, for example optimal stratification within predetermined industry classifications. In such a case any references to a population in this thesis can be equally applied to the relevant stage or subpopulation of a survey. Optimal stratification would however not be applicable if there are further constraints on the strata selected by optimal stratification, as this may produce perverse results.

1.6 Summary

This chapter has provided a brief overview of the foundations of optimal stratification, the importance of stratification in business and agricultural populations, and the scope of this thesis. The next chapter will cover some of the relevant foundations and theory associated with stratification, and chapter 3 will consider the problem of optimal allocation for stratified sampling.

Chapter 4 goes through the theory of optimal stratification, and provides the basis of work on optimal stratum boundaries. Chapters 5 to 8 concern various approximations for the optimal boundary points, covering the cumulative square root of frequency approach, Ekman's algorithm, the Lavallée-Hidiroglou algorithm, and some further approaches. Chapter 9 then looks at the problem of the optimal number of strata.

There is also a significant amount of code-based work undertaken in the construction of the various algorithms, and this is contained within the appendices. The appendices also include descriptions of the various populations used within this thesis.

Chapter 2

Background

2.1 Overview

Optimal stratification extends the usual concepts of stratification through providing algorithms for estimating the number of strata, the placement of stratum boundaries, and the allocation of sample units among the strata, in order to minimise the variance of estimates. However it is important to consider some of the foundations and properties of stratification before extending these concepts with such algorithms.

This chapter sets out the notation and framework relating to stratified sampling, and derives the standard equations that are used for the stratification of a population. We then examine one of the most common approaches, stratified random sampling, and consider the improvement in variance, and the "design effect", from a stratified random sampling design. Finally we look at some of the populations that will be used in the remainder of this thesis.

2.2 Notation

This thesis will consider stratification of a population y of size N at the boundary points of $k_0,\ k_1,\ \ldots,\ k_{L-1},\ k_L$ into L strata of size N_h. Each stratum h will have sample of size n_h, resulting in a total sample size of n.

The notation used in this thesis generally follows the notation used in sampling and stratification as follows (Rivest 2002):

y_{hi}	value of the ith unit in stratum h
$W_h = \dfrac{N_h}{N}$	stratum weight
$\bar{Y}_h = \dfrac{\sum_{i=1}^{N_h} y_{hi}}{N_h}$	population mean in stratum h
$T_h = \sum_{i=1}^{N_h} y_{hi} = N_h \bar{Y}_h$	population total in stratum h
$S_h^2 = \dfrac{\sum_{i=1}^{N_h} (y_{hi} - \bar{Y}_h)^2}{N_h - 1}$	population variance in stratum h

For stratified random sampling, these quantities become:

$f_h = \dfrac{n_h}{N_h}$	sampling fraction for stratum h
$\bar{y}_h = \dfrac{\sum_{i=1}^{n_h} y_{hi}}{n_h}$	sample mean in stratum h
$t_h = \dfrac{N_h}{n_h} \sum_{i=1}^{n_h} y_{hi} = N_h \bar{y}_h$	sample total in stratum h
$s_h^2 = \dfrac{\sum_{i=1}^{n_h} (y_{hi} - \bar{y}_h)^2}{n_h - 1}$	sample variance in stratum h

We consider the derivation of the overall population quantities T_{st}, $\bar{Y}_{st}$, $V(T_{st})$, and $V(\bar{Y}_{st})$ for the population total, population mean, variance of

the population total, and variance of the population mean respectively in the next section.

2.3 Stratification

2.3.1 Population Estimators

The *true* population total given the boundary points k_0, k_1, ..., k_{L-1}, k_L can be specified as:

$$T_{st} = \sum_{h=1}^{L} T_h \tag{2.1}$$

Likewise the true population mean using the above result, and $T_h = N_h \bar{Y}_h$ from the previous section, can be specified as:

$$\begin{aligned} \bar{Y}_{st} &= \frac{T_{st}}{N} \\ &= \frac{\sum_{h=1}^{L} T_h}{N} \\ &= \frac{\sum_{h=1}^{L} N_h \bar{Y}_h}{N} \\ &= \sum_{h=1}^{L} W_h \bar{Y}_h \end{aligned} \tag{2.2}$$

In both of the above, the true population total T_{st} and mean $\bar{Y}_h$ can be

expanded to give the standard equations for a population total:

$$\begin{aligned} T_{st} &= \sum_{h=1}^{L} T_h \\ &= \sum_{h=1}^{L} \sum_{i=1}^{N_h} y_{hi} \\ &= \sum_{i=1}^{N} y_i \end{aligned} \tag{2.3}$$

and population mean:

$$\begin{aligned} \bar{Y}_{st} &= \frac{T_{st}}{N} \\ &= \frac{\sum_{i=1}^{N} y_i}{N} \end{aligned} \tag{2.4}$$

These relations will be useful for evaluating the design effect of stratification in section 2.5.

The *estimated* population total for stratified sampling is:

$$\begin{aligned} t_{st} &= \sum_{h=1}^{L} t_h \\ &= \sum_{h=1}^{L} N_h \bar{y}_h \end{aligned} \tag{2.5}$$

using the relation $t_h = N_h \bar{y}_h$ from section 2.2. The estimated population

mean is:

$$\begin{aligned}\bar{y}_{st} &= \frac{\sum_{h=1}^{L} N_h \bar{y}_h}{N} \\ &= \sum_{h=1}^{L} W_h \bar{y}_h \end{aligned} \tag{2.6}$$

We can show that t_{st} is an unbiased estimate of the true population total T_{st} given in equation (2.1) as follows:

$$\begin{aligned} E\left(t_{st}\right) &= E\left(\sum_{h=1}^{L} t_h\right) \\ &= \sum_{h=1}^{L} E\left(t_h\right) \\ &= \sum_{h=1}^{L} T_h \end{aligned} \tag{2.7}$$

assuming the estimated stratum total t_h is an unbiased estimator of true stratum total T_h. Similarly $\bar{y}_{st}$ is an unbiased estimated of the true mean $\bar{Y}_{st}$ given in equation (2.2):

$$\begin{aligned} E\left(\bar{y}_{st}\right) &= E\left(\sum_{h=1}^{L} W_h \bar{y}_h\right) \\ &= \sum_{h=1}^{L} W_h E\left(\bar{y}_h\right) \\ &= \sum_{h=1}^{L} W_h \bar{Y}_h \end{aligned} \tag{2.8}$$

assuming the estimated stratum mean $\bar{y}_h$ is an unbiased estimator of true stratum mean $\bar{Y}_h$. Therefore t_{st} and $\bar{y}_{st}$ are unbiased estimators of the true

population total T_{st} and mean Y_{st} respectively.

2.3.2 Variance of the Estimators

The variance of the estimated population total t_{st} in equation (2.5) is:

$$\begin{aligned} V(t_{st}) &= V\left(\sum_{h=1}^{L} t_h\right) \\ &= \sum_{h=1}^{L} V(t_h) + 2\sum_{h=1}^{L}\sum_{j>h}^{L} Cov(t_h t_j) \end{aligned} \tag{2.9}$$

However each stratum is mutually exclusive, meaning the covariance between stratum estimators is equal to zero. Hence the variance of the estimated population total $V(t_{st})$ simply reduces to the sum of the individual stratum estimators:

$$V(t_{st}) = \sum_{h=1}^{L} V(t_h) \tag{2.10}$$

Likewise the variance of the estimated population mean $\bar{y}_{st}$ given in equation (2.6) is:

$$\begin{aligned} V(\bar{y}_{st}) &= V\left(\sum_{h=1}^{L} W_h \bar{y}_h\right) \\ &= \sum_{h=1}^{L} W_h^2 V(\bar{y}_h) + 2\sum_{h=1}^{L}\sum_{j>h}^{L} W_h W_j Cov(\bar{y}_h \bar{y}_j) \\ &= \sum_{h=1}^{L} W_h^2 V(\bar{y}_h) \end{aligned} \tag{2.11}$$

again with the covariance term being dropped as the covariance between the stratum estimators is equal to zero.

We can also derived the variance of the estimated population total $V(t_{st})$ using the relation $t_{st} = \sum_{h=1}^{L} N_h \bar{y}_h$ from equation (2.5) as follows:

$$\begin{aligned} V(t_{st}) &= V\left(\sum_{h=1}^{L} N_h \bar{y}_h\right) \\ &= \sum_{h=1}^{L} N_h^2 V(\bar{y}_h) + 2\sum_{h=1}^{L}\sum_{j>h}^{L} N_h N_j Cov(\bar{y}_h \bar{y}_j) \\ &= \sum_{h=1}^{L} N_h^2 V(\bar{y}_h) \\ &= N^2 \sum_{h=1}^{L} W_h^2 V(\bar{y}_h) \\ &= N^2 V(\bar{y}_{st}) \end{aligned} \tag{2.12}$$

This is useful in applying results relating to the variance of the estimated mean of a population $V(\bar{y}_{st})$ to the variance of the estimated total $V(t_{st})$.

The corresponding unbiased estimator of the variance of the estimated population total in (2.10) is:

$$\hat{V}(t_{st}) = \sum_{h=1}^{L} \hat{V}(t_h) \tag{2.13}$$

and the unbiased estimator of the variance of the estimated population mean in (2.11) is:

$$\hat{V}(\bar{y}_{st}) = \sum_{h=1}^{L} W_h^2 \hat{V}(\bar{y}_h) \tag{2.14}$$

We can show that $\hat{V}(t_{st})$ is an unbiased estimator of the variance of the estimated population total $V(t_{st})$ as follows:

$$\begin{aligned} E\left(\hat{V}(t_{st})\right) &= E\left(\sum_{h=1}^{L}\hat{V}(t_h)\right) \\ &= \sum_{h=1}^{L} E\left(\hat{V}(t_h)\right) \\ &= \sum_{h=1}^{L} V(t_h) \end{aligned} \tag{2.15}$$

assuming $\hat{V}(t_h)$ is an unbiased estimator of the variance of the estimated stratum total $V(t_h)$. Likewise $\hat{V}(\bar{y}_{st})$ is an unbiased estimator of the variance of the estimated population mean $V(\bar{y}_{st})$:

$$\begin{aligned} E\left(\hat{V}(\bar{y}_{st})\right) &= E\left(\sum_{h=1}^{L} W_h^2\hat{V}(\bar{y}_h)\right) \\ &= \sum_{h=1}^{L} W_h^2 E\left(\hat{V}(\bar{y}_h)\right) \\ &= \sum_{h=1}^{L} W_h^2 V(\bar{y}_h) \end{aligned} \tag{2.16}$$

again assuming $\hat{V}(\bar{y}_h)$ is an unbiased estimator of the variance of the estimated stratum mean $V(\bar{y}_h)$. Therefore $\hat{V}(t_{st})$ and $\hat{V}(\bar{y}_{st})$ are unbiased variance estimators of the estimated population total $V(t_h)$ and mean $V(\bar{y}_h)$.

This section has provided a brief introduction to the theory of stratification to obtain the estimated population mean and total, and shows the considerable interrelationship between the two estimators. In the interests of brevity, and consistent with the scope of the thesis, the following sections

and remainder of the thesis will be mainly concerned with results relating to the estimated population mean.

2.4 Stratified Random Sampling

If we employ a stratification design and take a simple random sample (without replacement) of size n_h from each stratum, then the overall sampling design is referred to as stratified random sampling. This section outlines the stratified random sampling approach, and will be the predominant approach used throughout this thesis.

Formulas for the estimated population total and estimated population mean for stratification have been previously given in equations (2.5) and (2.6) respectively. The results from the equations for the estimated stratum total t_h and stratum mean $\bar{y}_h$ in section 2.2 are then substituted into these formulas to provide overall estimates of the population total and population mean.

The variance of the stratum mean under stratified random sampling is given by:

$$V(\bar{y}_h) = \left(\frac{N_h - n_h}{N_h}\right)\frac{S_h^2}{n_h} \tag{2.17}$$

where $(N_h - n_h)/N_h$ is the finite population correction and S_h^2 is the variance in stratum h, as given in section 2.2. We substitute the above into equation

(2.12) to obtain the variance of the estimated total as follows:

$$\begin{aligned} V(t_{st}) &= \sum_{h=1}^{L} N_h^2 \left(\frac{N_h - n_h}{N_h} \right) \frac{S_h^2}{n_h} \\ &= \sum_{h=1}^{L} N_h \left(N_h - n_h \right) \frac{S_h^2}{n_h} \\ &= \sum_{h=1}^{L} N_h \left(\frac{1}{f_h} - 1 \right) S_h^2 \end{aligned} \tag{2.18}$$

Likewise we can substitute (2.17) into equation (2.11) to obtain the variance of the estimated mean:

$$\begin{aligned} V(\bar{y}_{st}) &= \sum_{h=1}^{L} W_h^2 \left(\frac{N_h - n_h}{N_h} \right) \frac{S_h^2}{n_h} \\ &= \sum_{h=1}^{L} W_h^2 \left(\frac{1}{n_h} - \frac{1}{N_h} \right) S_h^2 \\ &= \frac{1}{N} \sum_{h=1}^{L} W_h \left(\frac{1}{f_h} - 1 \right) S_h^2 \end{aligned} \tag{2.19}$$

The final forms of the above two equations are unusual, but convenient, variants for comparing optimal stratification algorithms on known (or enumerated) populations. Both of the final forms only depend on the population characteristics and the sampling fraction within each stratum, and therefore enable us to calculate the variance of the estimated total and mean under the various optimal stratification algorithms without actually needing to sample the relevant population. This mitigates the effect of any sample variation, and we use these results extensively in subsequent chapters to compare various optimal stratification algorithms.

An unbiased estimator of the stratum variance S_h^2 is the stratum sample variance s_h^2 as follows:

$$s_h^2 = \frac{1}{N_h - 1} \sum_{i=1}^{n_h} (y_{hi} - \bar{y}_h)^2 \tag{2.20}$$

Therefore an unbiased estimator of variance of the sample total using stratified random sampling is simply:

$$v(t_{st}) = s^2(t_{st}) = \sum_{h=1}^{L} N_h \left(N_h - n_h\right) \frac{s_h^2}{n_h} \tag{2.21}$$

and an unbiased estimator of the variance of the sample mean is:

$$\begin{aligned} v(\bar{y}_{st}) = s^2(\bar{y}_{st}) &= \sum_{h=1}^{L} \left(\frac{N_h}{N}\right)^2 \left(\frac{N_h - n_h}{N_h}\right) \frac{s_h^2}{n_h} \\ &= \sum_{h=1}^{L} \left(\frac{N_h}{N}\right)^2 \frac{s_h^2}{n_h} - \sum_{h=1}^{L} \left(\frac{N_h}{N}\right)^2 \left(\frac{n_h}{N_h}\right) \frac{s_h^2}{n_h} \\ &= \sum_{h=1}^{L} \frac{W_h^2 s_h^2}{n_h} - \sum_{h=1}^{L} \frac{W_h s_h^2}{N} \end{aligned} \tag{2.22}$$

The final form of the above equation can be convenient for computational purposes, as the last term represents the reduction in variance due to the finite population correction. The effect of the finite population correction is often dropped or ignored in the derivation of optimal stratification algorithms, and the above result is consequently used a number of times in subsequent chapters to derive and examine such algorithms.

When stratum sizes are sufficiently large (at least 30), we can construct an approximate $100(1-\alpha)\%$ confidence interval for the estimated population

total as follows (Lohr 1999):

$$t_{st} \pm z_{\alpha/2} s(t_{st}) \tag{2.23}$$

where multiplier $z_{\alpha/2}$ is from the normal distribution. Likewise we can construct a confidence interval for the estimated population mean:

$$\bar{y}_{st} \pm z_{\alpha/2} s(\bar{y}_{st}) \tag{2.24}$$

The t-distribution can be used if the stratum sample sizes are smaller than 30, with a formula for the approximate degrees of freedom (df) given by Satterthwaite (1946):

$$df = \frac{\left(\sum g_h s_h^2\right)^2}{\sum \frac{\left(g_h s_h^2\right)^2}{n_h - 1}} \tag{2.25}$$

where:

$$g_h = \frac{N_h(N_h - n_h)}{n_h} \tag{2.26}$$

Thompson (1992) and Cochran (1977) further discuss some of the issues in estimating confidence intervals for stratified random sampling; however the above is sufficient for our purposes.

2.5 Design Effect

The design effect (*deff*) of a survey design is usually specified as the ratio of variance of an estimate relative to the variance of an estimate from simple random sampling (Kish 1965). This can be formally stated for the variance of a mean from a stratified random sampling design as:

$$deff = \frac{V_{st}(\bar{y}_{st})}{V_{ran}(\bar{y}_{st})} \tag{2.27}$$

where $V_{st}(\bar{y}_{st})$ is the variance of the sample mean from stratified random sampling, and $V_{ran}(\bar{y}_{st})$ is the variance of the sample mean from simple random sampling. This provides a convenient measure to evaluate the effectiveness of a survey design, and to appraise the gain relative to other possible alternatives.

Unfortunately a sample obtained from stratification is not the same as a sample obtained from simple random sampling, meaning that we cannot simply use the standard formula for the variance of a mean for simple random sampling. Instead we need to derive a value for the variance for simple random sampling given the stratified sampling design. We use the estimator for the variance of a mean from simple random sampling using stratified random sampling given in (Rao 1962) as a basis of our derivation below.

We start with the usual unbiased estimator of the variance of a mean

from stratified random sampling:

$$\begin{aligned} V_{ran} &= \frac{(N-n)}{N}\frac{S^2}{n} \\ &= \frac{(N-n)}{nN}\frac{\sum_{i=1}^{N} y_i^2 - N\bar{Y}^2}{N-1} \\ &= \frac{(N-n)}{n(N-1)}\left(\frac{1}{N}\sum_{i=1}^{N} y_i^2 - \bar{Y}^2\right) \end{aligned} \tag{2.28}$$

The equation $\sum_{h=1}^{N} y_i^2$ can be restated in a manner similar to equation (2.3) as:

$$\sum_{h=1}^{L}\sum_{i=1}^{N_h} y_{hi}^2 = \sum_{i=1}^{N} y_i^2 \tag{2.29}$$

Substituting this into equation (2.28) gives:

$$V_{ran} = \frac{(N-n)}{n(N-1)}\left(\frac{1}{N}\sum_{h=1}^{L}\sum_{i=1}^{N_h} y_{hi}^2 - \bar{Y}^2\right) \tag{2.30}$$

We also notice that:

$$E\left(\frac{N_h}{n_h}\sum_{i=1}^{n_h} y_{hi}^2\right) = \sum_{i=1}^{N_h} y_{hi}^2 \tag{2.31}$$

and:

$$\begin{aligned} E\left(v(\bar{y}_{st})\right) &= V(\bar{y}_{st}) \\ &= E(\bar{y}_{st}^2) - E(\bar{y}_{st})^2 \\ &= E(\bar{y}_{st}^2) - \bar{Y}^2 \end{aligned} \tag{2.32}$$

as $v(\bar{y}_{st})$ is an unbiased estimator of $V(\bar{y}_{st})$, and $\bar{y}_{st}$ is an unbiased estimator of $\bar{Y}$. Rearranging this gives:

$$\bar{Y}^2 = \bar{y}_{st}^2 - v(\bar{y}_{st}) \tag{2.33}$$

Therefore substituting (2.31) and (2.33) into (2.30) gives the following unbiased estimator of the variance of the mean of a simple random sample $V_{ran}(\bar{y}_{st})$ from the same population as the stratified random sample:

$$V_{ran} = \frac{(N-n)}{n(N-1)} \left(\frac{1}{N} \sum_{h=1}^{L} \frac{N_h}{n_h} \sum_{i=1}^{n_h} y_{hi}^2 - \bar{y}_{st}^2 + v(\bar{y}_{st}) \right) \tag{2.34}$$

The above estimate of the variance from simple random sampling is implemented as part of the `summary.strata` algorithm in Appendix B.

It is interesting to note that if the allocation of sample units is proportional to stratum size, then Cochran (1977) shows that the equation in (2.34) reduces to:

$$\begin{aligned} V_{ran} &= \frac{(N-n)}{n(N-1)} \left(\frac{1}{n} \sum_{i=1}^{n} y_i^2 - \bar{y}_{st}^2 + v(\bar{y}_{st}) \right) \\ &= \frac{(N-n)}{n(N-1)} \left(\frac{n-1}{n} s^2 + v(\bar{y}_{st}) \right) \end{aligned} \tag{2.35}$$

When the overall sample size n is large, $(n-1) \approx n$ and $(N-1) \approx N$. The term in $v(\bar{y}_{st})$ is also of order $1/n$ relative to s^2, resulting in:

$$V_{ran} \approx \frac{(N-n)}{N} \frac{s^2}{n} \tag{2.36}$$

This therefore shows that the above equation for simple random sampling is only appropriate as an approximation of the variance of a simple random sample given a stratified random sampling design if the overall sample size is large and allocation of sample units among the strata is proportional to stratum size.

2.6 Populations

This thesis uses a number of populations to compare the results for various optimal stratification algorithms, and in particular the application of optimal boundary algorithms to highly skewed populations such as those present in business and agricultural populations. These populations are classified into three groups: "survey populations" with both survey (target population) and auxiliary information, "auxiliary variable populations" constituting only one variable (an auxiliary information variable assumed to be the same as the survey population), and "simulated populations" of survey and auxiliary information constructed by generating pairs of random numbers from a bivariate log-normal distribution.

This section provides a brief description of the populations used in this thesis, and further information and a description of the variables in these populations appears in Appendix A. Some of these populations represent only a "sample" of an overall larger population, however we will assume for the purposes of this thesis that they constitute the entire population of interest.

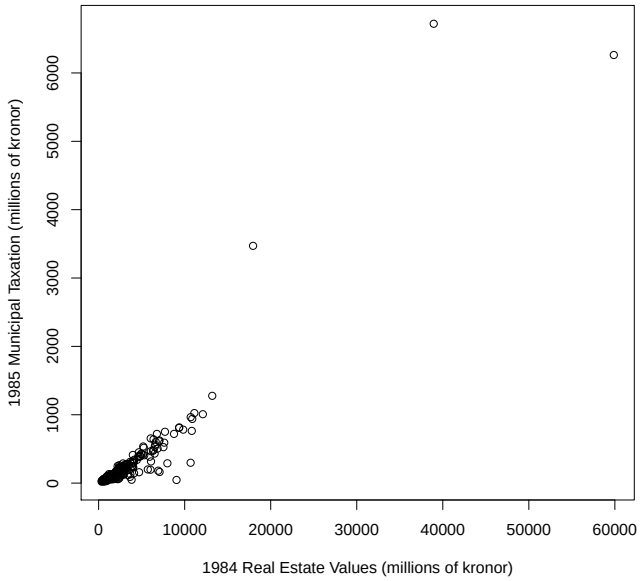

Figure 2.1: Scatter plot of 1985 Municipal Taxation (millions of kronor) and 1984 Real Estate Values (millions of kronor) from the Sweden Municipality population (MU284)

2.6.1 Survey Populations

The first of the survey populations is the Australian Agricultural and Grazing Industries Survey (AAGIS), and a scatter plot of this population appears as an example in section 1.4. We are primarily interested in the number of beef cattle, and stratify the population using the administrative farm area information. The overall population is highly skewed, and the correlation between the survey and auxiliary information is 0.75.

The scatter plot of the Sweden Municipality population (MU284) in figure 2.1 uses 1984 real estate values to estimate 1985 municipal taxation, and the scatter plot of the 2001 Survey of Household Spending (SHS) given in figure 2.2 uses the household income information to estimate values for

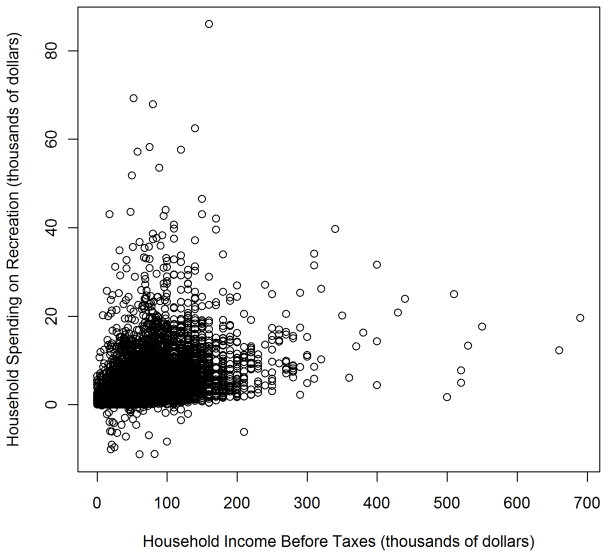

Figure 2.2: Scatter plot of Household Spending on Recreation (thousands of dollars) and Household Income Before Taxes (thousands of dollars) from the 2001 Survey of Household Spending (SHS)

household spending on recreation. The two variables from the MU284 population are highly correlated, with a correlation coefficient of 0.94, whereas the two variables from the SHS population have a correlation coefficient of just 0.50.

2.6.2 Auxiliary Variable Populations

The auxiliary variable populations used in this thesis are: the population of debtors used in Horgan (2003), the resources of large commercial US banks and the population of US cities in 1940 used in Cochran (1961), and a simulation of data from Monthly Retail Trade Survey (MRTS) of Statistics Canada used in Baillargeon, Rivest & Ferland (2007).

The histograms of debtors in figure 2.3 and the MRTS population in

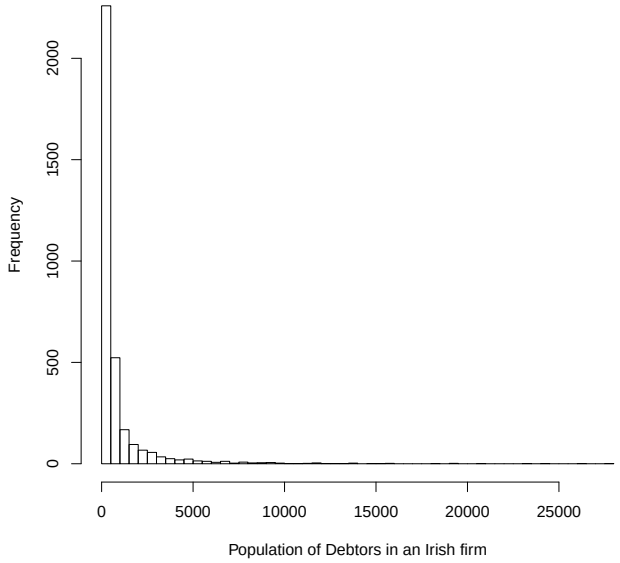

Figure 2.3: Histogram of a population of Debtors in an Irish firm

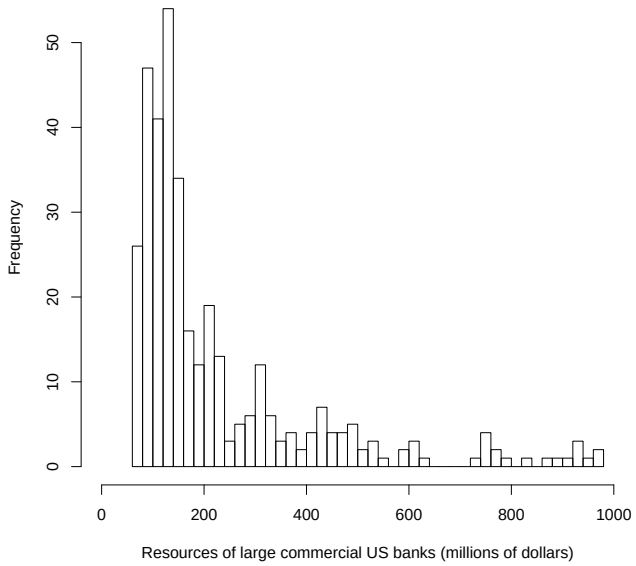

Figure 2.4: Histogram of the resources of large commercial US banks (millions of dollars)

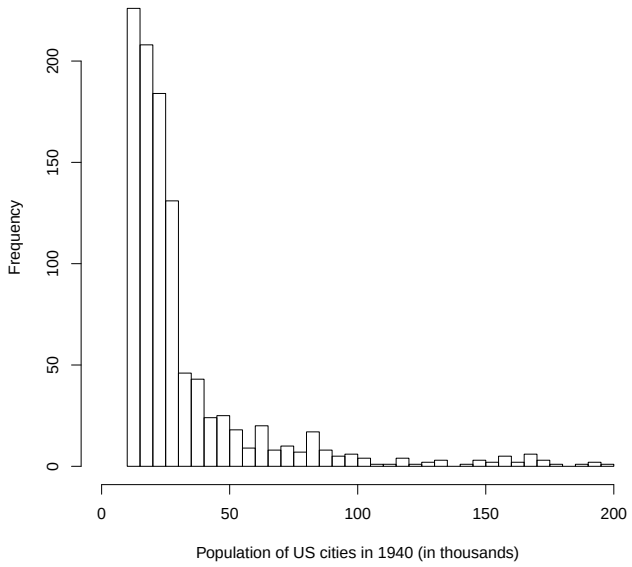

Figure 2.5: Histogram of the population of US cities in 1940 (in thousands)

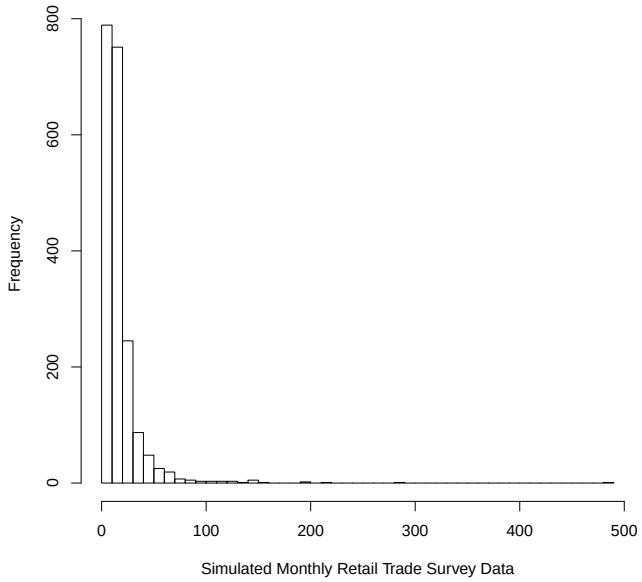

Figure 2.6: Histogram of simulated Monthly Retail Trade Survey Data (MRTS) of Statistics Canada

figure 2.6 are both highly skewed, each with concentrations of values around or near zero. The US banks in figure 2.4 and US cities in figure 2.5 are less skewed, each with a considerable number of larger values.

2.6.3 Simulated Populations

Many of the above populations, and indeed many business and agricultural populations, can be approximately represented by or have similar characteristics to a log-normal distribution (Hedlin 2003). We can therefore consider simulating business and agricultural populations by generating pairs of random numbers from a bivariate log-normal distribution to represent the survey and auxiliary information for such populations.

We can generate random numbers for a log-normal distribution by simply taking the exponential of random numbers from the corresponding normal distribution. However the variance-covariance matrix in the generation of bivariate log-normal random numbers is that of the underlying normal distribution, and requires some manipulation in order to directly specify the correlation between the resulting log-normal survey population and auxiliary information.

The correlation of variables Y_i and Y_j from a bivariate log-normal distribution is given in Johnson & Kotz (1972) as follows:

$$\text{corr}(Y_i, Y_j) = \frac{(\exp(\rho_{ij}\sigma_i\sigma_j) - 1)}{\sqrt{(\exp(\sigma_i^2) - 1)\left(\exp(\sigma_j^2) - 1\right)}} \tag{2.37}$$

where $\rho_{ij} = corr(Z_i, Z_j)$, $Z_i = \log(Y_i)$, and $Z_j = \log(Y_j)$. We notice that

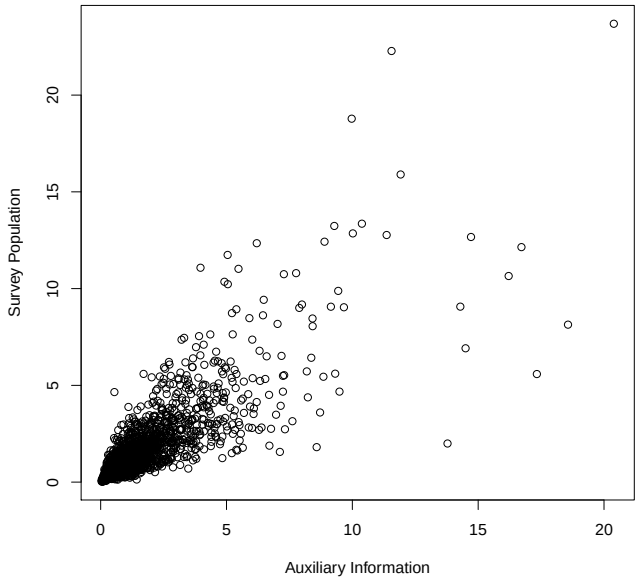

Figure 2.7: Scatter plot of Survey Population and Auxiliary Information for a Simulated Bivariate Log-Normal Population ($N = 2000$)

that the covariance of the normal random variables Z_i and Z_j is simply:

$$\begin{aligned} \text{cov}(Z_i, Z_j) &= \frac{\text{corr}(Z_i, Z_j)}{\sigma_i \sigma_j} \\ &= \frac{\rho_{ij}}{\sigma_i \sigma_j} \end{aligned} \tag{2.38}$$

and hence equation (2.37) can be restated as:

$$\text{cov}(Z_i, Z_j) = \log\left(\text{corr}(Y_i, Y_j)\sqrt{(\exp(\sigma_i^2) - 1)\left(\exp(\sigma_j^2) - 1\right)} + 1\right) \tag{2.39}$$

This therefore gives the covariance ($\text{cov}(Z_i,\ Z_j)$) of the bivariate normal distribution to be used in the generation of bivariate log-normal random variables, in order to generate a population with a given correlation ($\text{corr}(Y_i,\ Y_j)$) between the survey and auxiliary information.

We will primarily use a correlation coefficient of 0.8 and a population size of $N = 2000$ in any simulation of survey populations and auxiliary information in this thesis (unless stated otherwise). We also specify values for the mean and standard deviation of the logarithm of zero and one respectively, as this results in a population with similar characteristics to many of the other populations encountered in this thesis and elsewhere, and give an example of such a simulated bivariate log-normal population in figure 2.7.

The results from the above equations for the covariance of a bivariate normal distribution are also implemented as part of the `mvrlnorm` function in Appendix B to generate multivariate log-normal random variables, and enables direct specification of the correlation between the simulated survey population and auxiliary information through the use of a "correlation matrix" parameter.

2.7 Summary

This chapter has outlined the notation and basic theory for stratification, and in particular has covered several of the important results for stratified random sampling in order to estimate the mean and total of a population. We have also considered the use of population values to calculate the variance of the estimated mean and total in order to minimise the effect of any sample variation on estimators (or comparisons between the various optimal stratification algorithms), and have derived formula for the variance of estimators from a simple random sample given the results from a stratified random sample design.

We have looked at eight populations that will be used in the remainder of the thesis: three "survey populations" (of survey and auxiliary information), four univariate "auxiliary variable populations" (comprised of an auxiliary variable assumed to be the same as the survey variable), and one simulated population. The simulated population generates pairs of random numbers from a bivariate log-normal distribution, and includes a derivation of the correlation matrix in order to directly specify the correlation between the survey population and auxiliary information (which is otherwise set at 0.8 for the purposes of this thesis).

The following chapter will add to the theory covered in this chapter by outlining algorithms for the allocation of sampling units to the various strata. The results of these two chapters will then be used in the subsequent chapters in order to derive algorithms for the construction of optimal stratum boundaries.

Chapter 3

Allocation

3.1 Overview

The allocation of sampling units among the strata usually occurs after the construction of strata, as one of the final stages in stratified sampling design. However most of the algorithms for the construction of optimal stratum boundaries assume some form of allocation in order to derive optimal boundary algorithms. This therefore makes in pertinent to consider the allocation of sampling units prior to the construction of stratum boundaries in order to be able to build on this theory in subsequent chapters.

One of the simplest allocation schema is to select units from each stratum using a sample size that is proportional to the overall population size in each stratum. This is known as proportional allocation, and we have already referred to such a strategy in section 2.5. However proportional allocation is rarely used in business surveys as there is usually more variation in the values from larger organisations compared to smaller businesses, and is generally

restricted to situations where there is little information regarding variation and sampling costs of population values (Sigman & Monsour 1995).

This chapter investigates optimal allocation strategies, in order to derive minimum variance or minimum cost estimates. Optimal allocation assumes that there is information available on the variation in population values and the cost of sampling from different strata, and takes advantage of this information in order to improve the variance or cost of sample estimates. We then consider the relative efficiency of the various allocation strategies, the increase in variance from approximately optimal allocation, and the special case of "take-all" strata.

3.2 Optimal Allocation

Optimal allocation is concerned with the minimisation of the variance (V) of an estimator (such as $\bar{y}_{st}$) for a given cost and sample size, or minimising the cost (C) of the estimator for a given level of variance. Both derivations have analogous results, particularly when using stratified random sampling.

We first propose a modified derivation of the stratum sample size for a given total sample size, using Lagrange multipliers, which facilitates examination of the rate of change in the optimal stratum sample size for changes the variance or cost of values within a stratum. We then compare this with the standard derivation using the Cauchy-Schwarz inequality, and use these results to derive a general formulation for stratum sample size for minimum variance or minimum cost estimators. Finally we apply these results to the estimated variance of the sample mean from stratified random sampling given

in section 2.4.

To find stratum sample size n_h for optimal allocation, we need to assume some functional form for the variance and cost of sample units from strata. We first consider a general class of estimators with variance:

$$V = V_0 + V' = V_0 + \sum_{h=1}^{L} \frac{V_h^2}{n_h} \tag{3.1}$$

where V_0 is the fixed component of the estimator's variance, and V_1, ..., V_L are constants relating to the variance of units in each stratum. This includes a large set of estimators, and importantly encompasses the estimators thus far considered (Kish 1976).

We also approximate the total survey cost using the function C of the form:

$$C = C_0 + C' = C_0 + \sum_{h=1}^{L} C_h n_h \tag{3.2}$$

where C_0 is the fixed cost of the survey, C' is the variable costs, and C_1, ..., C_L are constants relating to the variable cost of sampling from each stratum. This assumes that the cost of the sample from each stratum is proportional to the stratum size, and the cost per unit sampled does not vary within each stratum.

We set up the Lagrange multiplier for the calculation of the stratum sample sizes n_h using:

$$\Lambda(n_h, \lambda) = f(n_h) + \lambda(g(n_h) - c) \tag{3.3}$$

where $g(n_h) = c$. We substitute in the variance and cost functions into the above:

$$\Lambda(n_h, \lambda) = V_0 + \sum_{h=1}^{L} \frac{V_h^2}{n_h} + \lambda \left(C_0 - \sum_{h=1}^{L} C_h n_h - C \right) \tag{3.4}$$

and calculate the partial derivate with respect to n_h:

$$\frac{\partial \Lambda}{\partial n_h} = \frac{V_h^2}{n_h} + \lambda C_h n_h^2 = 0 \tag{3.5}$$

for $n_h = \{n_0, \ldots, n_L\}$. Solving for n_h results in:

$$n_h^2 = \frac{1}{\lambda} \frac{V_h^2}{C_h} \tag{3.6}$$

or:

$$n_h = \frac{1}{\sqrt{\lambda}} \frac{V_h}{\sqrt{C_h}} \tag{3.7}$$

where $1/\sqrt{\lambda}$ is the rate of change in n_h from a change in $V_h/\sqrt{C_h}$. Alternatively we can state:

$$n_h \propto \frac{V_h}{\sqrt{C_h}} \tag{3.8}$$

The above derivation is equivalent to minimisation of product $(V'C')$ of the "variable" components of the variance V' and cost C' functions using the Cauchy-Schwarz inequality. The process using the Cauchy-Schwarz inequality derives the stratum sample sizes using equations (3.1) and (3.2) as

follows (Stuart 1954):

$$V'C' = \left(\sum_{h=1}^{L} \frac{V_h^2}{n_h}\right)\left(\sum_{h=1}^{L} C_h n_h\right) \tag{3.9}$$

The Cauchy-Schwarz inequality states for two sets of positive numbers a_h and b_h that:

$$\left(\sum_{h=1}^{L} a_h^2\right)\left(\sum_{h=1}^{L} b_h^2\right) \geq \left(\sum_{h=1}^{L} a_h b_h\right)^2 \tag{3.10}$$

with the equality occurring if and only if a_h/b_h is constant for all strata. If we use V' from (3.1) and set:

$$a_h = \sqrt{\frac{V_h^2}{n_h}} = \frac{V_h}{\sqrt{n_h}} \tag{3.11}$$

and C' from (3.2) to set:

$$b_h = \sqrt{C_h n_h} \tag{3.12}$$

the inequality then becomes:

$$V'C' = \left(\sum_{h=1}^{L} \frac{V_h^2}{n_h}\right)\left(\sum_{h=1}^{L} C_h n_h\right) \geq \left(\sum_{h=1}^{L} V_h \sqrt{C_h}\right)^2 \tag{3.13}$$

Therefore the minimum value of $V'C'$ occurs when:

$$\frac{a_h}{b_h} = \frac{V_h}{n_h \sqrt{C_h}} \tag{3.14}$$

is constant. This is equivalent to the result in (3.8) using Lagrange multipliers where:

$$\lambda = \frac{a_h}{b_h} \tag{3.15}$$

Both results can then be constructed in relation to the total sample size n as:

$$n_h = \frac{V_h/\sqrt{C_h}}{\sum_{i=1}^{L} V_i/\sqrt{C_i}} n \tag{3.16}$$

giving the stratum sample sizes n_h for a given sample size n.

We now use this to construct the minimum variance and minimum cost estimators of n and n_h. The minimum variance estimator of n for a given total cost C can be constructed by first restating (3.2) as follows:

$$C - C_0 = \sum_{h=1}^{L} C_h n_h \tag{3.17}$$

We then substitute in the value of n_h from equation (3.16):

$$\begin{aligned} C - C_0 &= \sum_{h=1}^{L} C_h \frac{V_h/\sqrt{C_h}}{\sum_{i=1}^{L} V_i/\sqrt{C_i}} n \\ &= \frac{\sum_{h=1}^{L} V_h\sqrt{C_h}}{\sum_{i=1}^{L} V_i/\sqrt{C_i}} n \end{aligned} \tag{3.18}$$

Rearranging this for n gives:

$$n = \frac{\sum_{i=1}^{L} V_i/\sqrt{C_i}}{\sum_{h=1}^{L} V_h\sqrt{C_h}} (C - C_0) \tag{3.19}$$

This is the total sample size for the minimum variance estimator given a total cost C. We can then calculate the stratum sample sizes by substituting the above back into equation (3.16):

$$\begin{aligned} n_h &= \frac{V_h/\sqrt{C_h}}{\sum_{i=1}^{L} V_i/\sqrt{C_i}} \frac{\sum_{i=1}^{L} V_i/\sqrt{C_i}}{\sum_{j=1}^{L} V_j\sqrt{C_j}}(C - C_0) \\ &= \left(\frac{V_h}{\sqrt{C_h}}\right)\left(\frac{C - C_0}{\sum_{j=1}^{L} V_j\sqrt{C_j}}\right) \end{aligned} \tag{3.20}$$

This therefore gives the stratum sample size n_h for the minimum variance estimator for a given total cost C.

We can similarly derive the minimum cost estimator for a given variance by rewriting (3.1) as follows:

$$V - V_0 = \sum_{h=1}^{L} \frac{V_h^2}{n_h} \tag{3.21}$$

Substituting in n_h from equation (3.16):

$$\begin{aligned} V - V_0 &= \sum_{h=1}^{L} \frac{V_h^2}{n} \frac{\sum_{i=1}^{L} V_i/\sqrt{C_i}}{V_h/\sqrt{C_h}} \\ &= \sum_{h=1}^{L} \frac{V_h\sqrt{C_h}}{n} \sum_{i=1}^{L} V_i/\sqrt{C_i} \end{aligned} \tag{3.22}$$

Again rearranging for n gives:

$$n = \frac{\sum_{h=1}^{L} V_h\sqrt{C_h}}{V - V_0} \sum_{i=1}^{L} V_i/\sqrt{C_i} \tag{3.23}$$

which is the total sample size for the minimum cost estimator given a total

cost V. Substituting this back into equation (3.16):

$$\begin{aligned} n_h &= \frac{V_h/\sqrt{C_h}}{\sum_{i=1}^{L} V_i/\sqrt{C_i}} \frac{\sum_{j=1}^{L} V_j\sqrt{C_j}}{V - V_0} \sum_{i=1}^{L} V_i/\sqrt{C_i} \\ &= \left(\frac{V_h}{\sqrt{C_h}}\right)\left(\frac{\sum_{j=1}^{L} V_h\sqrt{C_j}}{V - V_0}\right) \end{aligned} \tag{3.24}$$

This therefore gives the stratum sample size n_h for the minimum cost estimator for a given total cost V, in a similar manner to the minimum variance estimator for a given total cost C in equation (3.20).

If the costs associated with sampling are unknown, or the fixed costs C_0 are estimated as zero and the variable costs C_h are constant, then the equations in (3.16) and (3.20) both reduce to:

$$n_h = \frac{V_h}{\sum_{i=1}^{L} V_i} n \tag{3.25}$$

This is often referred to as Neyman allocation, due to the work on the result in Neyman (1934). This will be particularly important in later work on stratum boundaries, as most of the algorithms ignore any effect from differential costs in sampling from strata.

The estimated variance of the sample mean for stratified random sampling was given in (2.22) of section 2.4 as follows:

$$v(\bar{y}_{st}) = \sum_{h=1}^{L} \frac{W_h^2 s_h^2}{n_h} - \sum_{h=1}^{L} \frac{W_h s_h^2}{N} \tag{3.26}$$

The above formula for the variance is of the form given in equation (3.1),

where:

$$V_h = W_h s_h \tag{3.27}$$

and:

$$V_0 = -\sum_{h=1}^{L} \frac{W_h s_h^2}{N} \tag{3.28}$$

The last of these terms is the finite population correction, and is consequently not affected by changes in stratum sample sizes. If we assume a cost function of the form given in equation (3.2) as follows:

$$C = c_0 + \sum_{h=1}^{L} c_h n_h \tag{3.29}$$

then the optimal stratum sample size for a given total sample size from equation (3.16) becomes:

$$\begin{aligned} n_h &= \frac{W_h s_h / \sqrt{c_h}}{\sum_{i=1}^{L} W_i s_i / \sqrt{c_i}} n \\ &= \frac{N_h s_h / \sqrt{c_h}}{\sum_{i=1}^{L} N_i s_i / \sqrt{c_i}} n \end{aligned} \tag{3.30}$$

The minimum variance estimator for a given cost c is obtained by substituting

equation (3.29) into equation (3.20) as follows:

$$\begin{aligned} n_h &= \left(\frac{W_h s_h}{\sqrt{c_h}}\right)\left(\frac{C - C_0}{\sum_{j=1}^{L} W_j s_j \sqrt{c_j}}\right) \\ &= \left(\frac{N_h s_h}{\sqrt{c_h}}\right)\left(\frac{C - c_0}{\sum_{j=1}^{L} N_j s_j \sqrt{c_j}}\right) \end{aligned} \tag{3.31}$$

Likewise the minimum cost estimator for a given variance $v(\bar{y}_{st})$ is obtained by substituting equations (3.27) and (3.28) into equation (3.24):

$$n_h = \left(\frac{W_h s_h}{\sqrt{c_h}}\right)\left(\frac{\sum_{j=1}^{L} W_j s_j \sqrt{c_j}}{V + (1/N)\sum_{h=1}^{L} W_h s_h^2}\right) \tag{3.32}$$

As mentioned previously, much of the work on optimal stratum boundaries assume Neyman allocation. This results in a stratum sample size n_h for a fixed total sample size of:

$$n_h = \frac{W_h s_h}{\sum_{i=1}^{L} W_i s_i} n \tag{3.33}$$

We can derive the formula for the minimum variance estimator for Neyman allocation by substituting equation (3.33) into (3.26) as follows:

$$\begin{aligned} v(\bar{y}_{st}) &= \sum_{h=1}^{L} W_h^2 s_h^2 \frac{\sum_{i=1}^{L} W_i s_i}{W_h s_h n} - \sum_{h=1}^{L} \frac{W_h s_h^2}{N} \\ &= \sum_{h=1}^{L} W_h s_h \frac{\sum_{i=1}^{L} W_i s_i}{n} - \sum_{h=1}^{L} \frac{W_h s_h^2}{N} \\ &= \frac{\left(\sum_{h=1}^{L} W_h s_h\right)^2}{n} - \frac{\sum_{h=1}^{L} W_h s_h^2}{N} \end{aligned} \tag{3.34}$$

This forms part of the core work of Dalenius (1950) that will be covered in chapter 4.

We can furthermore derive the stratum sample size n_h using proportional allocation for a fixed total size n by setting s_h to a constant value:

$$\begin{aligned} n_h &= \frac{W_h}{\sum_{i=1}^{L} W_i} n \\ &= \frac{N_h}{N} n \\ &= W_h n \end{aligned} \tag{3.35}$$

The formula for the variance of the sample mean using proportional allocation can then be derived by substituting (3.35) into (3.26):

$$\begin{aligned} v(\bar{y}_{st}) &= \sum_{h=1}^{L} \frac{W_h^2 s_h^2}{W_h n} - \sum_{h=1}^{L} \frac{W_h s_h^2}{N} \\ &= \frac{\sum_{h=1}^{L} W_h s_h^2}{n} - \frac{\sum_{h=1}^{L} W_h s_h^2}{N} \end{aligned} \tag{3.36}$$

For proportional allocation, $f_h = n_h/N_h = n/N$, meaning (3.36) can be reduced to:

$$\begin{aligned} v(\bar{y}_{st}) &= \frac{N-n}{Nn} \sum_{h=1}^{L} W_h s_h^2 \\ &= \frac{1-f_h}{n} \sum_{h=1}^{L} W_h s_h^2 \end{aligned} \tag{3.37}$$

The above results will be convenient in comparing the different allocation strategies in the next section.

3.3 Comparison of Allocation Strategies

Optimal allocation should result in an improvement (or decrease) in the variance of estimators compared to proportional allocation. Likewise proportional allocation should represent an improvement from simple random sampling. This section compares the estimated variance of a mean under optimal and proportional allocation with simple random sampling using a fixed total sample size in order to find the improvement in variance from these allocation strategies.

The formula for the estimated variance of the sample mean for stratified random sampling using optimal allocation is given in equation (3.34) as:

$$V_{opt} = \frac{\left(\sum_{h=1}^{L} W_h S_h\right)^2}{n} - \frac{\sum_{h=1}^{L} W_h S_h^2}{N} \tag{3.38}$$

and the formula for proportional allocation from equation (3.36) is

$$V_{prop} = \frac{\sum_{h=1}^{L} W_h S_h^2}{n} - \frac{\sum_{h=1}^{L} W_h S_h^2}{N} \tag{3.39}$$

The improvement in variance due to optimal allocation can therefore be

Table 3.1: Analysis of variance for stratified random sampling

Source	*df*	Sum of squares
Between strata	$L-1$	$\sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2 = \sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2$
Within strata	$N-L$	$\sum_{h=1}^{L}\sum_{i=1}^{L}(y_{hi} - \bar{Y}_h)^2 = \sum_{h=1}^{L}(N_h - 1)S_h^2$
Total	$N-1$	$\sum_{h=1}^{L}\sum_{i=1}^{L}(y_{hi} - \bar{Y})^2 = (N-1)S^2$

calculated as follows:

$$\begin{aligned} V_{prop} - V_{opt} &= \frac{\sum_{h=1}^{L} W_h S_h^2}{n} - \frac{\left(\sum_{h=1}^{L} W_h S_h\right)^2}{n} \\ &= \frac{1}{n}\left(\sum_{h=1}^{L} W_h S_h^2 - \left(\sum_{h=1}^{L} W_h S_h\right)^2\right) \\ &= \frac{1}{n}\left(\sum_{h=1}^{L} W_h (S_h - \bar{S})^2\right) \end{aligned} \tag{3.40}$$

where $\bar{S} = \sum_{h=1}^{L} W_h S_h$ is a weighted mean of the S_h (Cochran 1977). Therefore as $\bar{S}$ is less than S_h, by definition $V_{opt} \leq V_{prop}$.

The variance for the sample mean from simple random sampling is:

$$V_{ran} = (1-f)\frac{S^2}{n} \tag{3.41}$$

This can be restated by following the process described in Lohr (1999) using

the analysis of variance identities given in table 3.1:

$$\begin{aligned} V_{ran} &= \frac{(1-f)}{n(N-1)}(N-1)S^2 \\ &= \frac{(1-f)}{n(N-1)}\left(\sum_{h=1}^{L}\sum_{i=1}^{L}(y_{hi}-\bar{Y})^2\right) \\ &= \frac{(1-f)}{n(N-1)}\left(\sum_{h=1}^{L}(N_h-1)S_h^2+\sum_{h=1}^{L}N_h(\bar{Y}_h-\bar{Y})^2\right) \end{aligned} \tag{3.42}$$

We can rearrange the first term inside the brackets to obtain:

$$\begin{aligned} \sum_{h=1}^{L}(N_h-1)S_h^2 &= \frac{1}{N}\sum_{h=1}^{L}N(N_h-1)S_h^2 \\ &= \frac{1}{N}\sum_{h=1}^{L}\left(NN_hS_h^2-NS_h^2+N_hS_h^2-N_hS_h^2\right) \\ &= \frac{1}{N}\sum_{h=1}^{L}\left((N-1)N_hS_h^2-(N-N_h)S_h^2\right) \\ &= (N-1)\sum_{h=1}^{L}W_hS_h^2-\sum_{h=1}^{L}(1-W_h)S_h^2 \end{aligned} \tag{3.43}$$

Substituting this back into equation (3.42) gives:

$$\begin{aligned} V_{ran} &= \frac{(1-f)}{n(N-1)}\left((N-1)\sum_{h=1}^{L}W_hS_h^2+\sum_{h=1}^{L}(1-W_h)S_h^2-\sum_{h=1}^{L}N_h(\bar{Y}_h-\bar{Y})^2\right) \\ &= \frac{(1-f)}{n}\sum_{h=1}^{L}W_hS_h^2+\frac{(1-f)}{n(N-1)}\left(\sum_{h=1}^{L}N_h(\bar{Y}_h-\bar{Y})^2-\sum_{h=1}^{L}(1-W_h)S_h^2\right) \\ &= V_{prop}+\frac{(1-f)}{n(N-1)}\left(\sum_{h=1}^{L}N_h(\bar{Y}_h-\bar{Y})^2-\sum_{h=1}^{L}(1-W_h)S_h^2\right) \end{aligned} \tag{3.44}$$

Therefore proportional allocation provides better results that simple random

sampling unless:

$$\sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2 < \sum_{h=1}^{L}(1 - W_h)S_h^2 \tag{3.45}$$

This will hold as long as the values of N_h are large, and hence $N_h(\bar{Y}_h - \bar{Y})^2 > S_h^2$.

When the values of $1/N_h$, and hence $1/N$, are negligible, the formula in equation (3.42) becomes:

$$\begin{aligned} V_{ran} &= \frac{(1-f)}{n(N-1)}\left(\sum_{h=1}^{L}(N_h - 1)S_h^2 + \sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2\right) \\ &= \frac{(1-f)}{nN}\left(\sum_{h=1}^{L} N_h S_h^2 + \sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2\right) \\ &= \frac{(1-f)}{n}\sum_{h=1}^{L} W_h S_h^2 + \frac{(1-f)}{Nn}\sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2 \\ &= V_{prop} + \frac{(1-f)}{Nn}\sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2 \end{aligned} \tag{3.46}$$

where the term on the right represents the between strata sum of squares. Substituting in the results from equations (3.40) into (3.46) above gives:

$$V_{ran} = V_{opt} + \frac{1}{n}\sum_{h=1}^{L} W_h(S_h - \bar{S})^2 + \frac{(1-f)}{n}\sum_{h=1}^{L} W_h(\bar{Y}_h - \bar{Y})^2 \tag{3.47}$$

This shows that the decrease in variance from optimal allocation is first from accounting for the between strata sum of squares $W_h(\bar{Y}_h - \bar{Y})^2$ (resulting in proportional allocation), and then from the differences among the stratum standard deviations $W_h(S_h - \bar{S})^2$. Therefore equations (3.40), (3.46), and

(3.47) show that:

$$V_{opt} \leq V_{prop} \leq V_{ran} \tag{3.48}$$

where the values of $1/N_h$ are negligible.

If the values of $1/N_h$ are not negligible, then equation (3.45) can be restated in term of Neyman allocation where all $S_h^2 = S_w^2$ as follows (Cochran 1977):

$$\sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2 < (L-1)S_w^2 \tag{3.49}$$

Rearranging this becomes:

$$\frac{\sum_{h=1}^{L} N_h(\bar{Y}_h - \bar{Y})^2}{L-1} < S_w^2 \tag{3.50}$$

This implies that Neyman allocation will result in an increase in variance relative to simple random sampling when the mean square among strata is smaller than the mean square within strata (or the F-ratio is less than 1).

3.4 Proximal Allocation

The estimated variance of the population in each stratum can be imprecise, and the sample allocation for each stratum may only approximate the optimal allocation. At the very least, the sample allocation for each stratum has to be an integer whereas the values from the equations of n_h from equation (3.33) for optimal allocation may not be integers.

Unfortunately this can mean that the benefits from optimal allocation can be exaggerated, and the simplicity of strategies such as the self-weighting proportional allocation may be worth a 10% to 20% increase in variance (Sigman & Monsour 1995). We therefore turn our attention in this section to the relative loss or increase in variance from a proximal allocation, and review some of the work on deviations from the optimal allocation given in Cochran (1977).

The minimum variance equation (3.34) from section 3.2 is:

$$V(\bar{y}_{st}) = \frac{\left(\sum_{h=1}^{L} W_h S_h\right)^2}{n} - \frac{\sum_{h=1}^{L} W_h S_h^2}{N} \tag{3.51}$$

However the variance V' for the actual stratum sample size n'_h using equation (3.26) is:

$$V'(\bar{y}_{st}) = \sum_{h=1}^{L} \frac{W_h^2 S_h^2}{n'_h} - \sum_{h=1}^{L} \frac{W_h S_h^2}{N} \tag{3.52}$$

We can therefore calculate the increase in variance from the proximal allocation as follows:

$$V'(\bar{y}_{st}) - V(\bar{y}_{st}) = \sum_{h=1}^{L} \frac{W_h^2 S_h^2}{n'_h} - \frac{\left(\sum_{h=1}^{L} W_h S_h\right)^2}{n} \tag{3.53}$$

Substituting in $W_h S_h = n_h \sum_{i=1}^{L} W_i S_i / n$ from equation (3.26) into (3.53)

gives:

$$V'(\bar{y}_{st}) - V(\bar{y}_{st}) = n_h^2 \sum_{h=1}^{L} \frac{\left(\sum_{i=1}^{L} W_i S_i\right)^2}{n'_h n^2} - \frac{n\left(\sum_{h=1}^{L} W_h S_h\right)^2}{n^2}$$
$$= \frac{\left(\sum_{i=1}^{L} W_i S_i\right)^2}{n^2} \left(\sum_{h=1}^{L} \frac{n_h^2}{n'_h} - n\right)$$
$$= \frac{\left(\sum_{i=1}^{L} W_i S_i\right)^2}{n^2} \sum \frac{(n'_h - n_h)^2}{n'_h} \tag{3.54}$$

If we assume that the finite population correction is negligible, then equation (3.51) reduces to:

$$V(\bar{y}_{st}) = \frac{\left(\sum_{h=1}^{L} W_h S_h\right)^2}{n} \tag{3.55}$$

and:

$$\frac{V'(\bar{y}_{st}) - V(\bar{y}_{st})}{V(\bar{y}_{st})} = \frac{1}{n} \sum_{h=1}^{L} \frac{(n'_h - n_h)^2}{n'_h} \tag{3.56}$$

We can now calculate the proximal loss using the results of Kish (1976) as follows:

$$L = \frac{V'(\bar{y}_{st}) - V(\bar{y}_{st})}{V(\bar{y}_{st})} = \sum_{h=1}^{L} \frac{\hat{n}_h}{n} g_h^2 \tag{3.57}$$

where $g_h = n_h - n'_h/n'_h$ is the relative difference of the optimal allocation from the actual allocation. This therefore shows the increase in variance from a proximal allocation compared to the optimal allocation considered in

previous sections.

3.5 Take-All Strata

Stratification of highly skewed populations, such as those often observed in business and agricultural surveys, can result in some stratum populations that are considerably more variable than the populations in other strata. The application of optimal or Neyman allocation to such populations often produces a sampling fraction $f_h = n_h/N_h$, using the equations in section 3.2, that may be greater than one. This generally results in the construction of one or more "take-all" strata, whereby all population values are sampled within the stratum.

Several adjustments are required to the equations for the "take-some" strata that we have implicitly assumed in previous sections in order to accommodate the new take-all stratum (or strata). If we denote the take-some strata as the set $\{1, \ldots, J\}$ and the take-all strata as $\{J+1, \ldots, L\}$, then we set:

$$n_j = N_j \tag{3.58}$$

for $j = \{J+1, \ldots, L\}$. The stratum sample size n_h for the take-some strata can then be specified as:

$$n_h = (n - \sum_{j=J+1}^{L} N_j)a_h \tag{3.59}$$

where a_h is the sample allocation strategy. The corresponding total sample

size n' of the take-some strata is then given by:

$$n' = \sum_{h=1}^{J} n_h \tag{3.60}$$

The take-some stratum sample size for stratified random sampling with Neyman allocation given in equation (3.33) can now be revised to incorporate the possibility of take-all strata as follows:

$$n_h = (n - \sum_{j=J+1}^{L} N_j) \frac{W_h s_h}{\sum_{i=1}^{J} W_i s_i} \tag{3.61}$$

and the formula for the estimated variance of the sample mean for stratified random sampling given in equation (3.34) of section 3.2 becomes:

$$v(\bar{y}_{st}) = \frac{\left(\sum_{h=1}^{J} W_h s_h\right)^2}{n'} - \frac{\sum_{h=1}^{J} W_h s_h^2}{N} \tag{3.62}$$

The above formula is also incorporated as part of the variance calculations of the `summary.strata` function in Appendix B.

The most common scenario is to construct a single take-all stratum, which results in the simple case of $n_L = N_L$. The stratum sample size calculation for the take some strata n'_h now reduces to:

$$n'_h = (n - N_L) a_h \tag{3.63}$$

and the stratum sample size for stratified random sampling with Neyman

allocation is:

$$n'_h = (n - N_L)\frac{W_h s_h}{\sum_{i=1}^{L-1} W_i s_i} \tag{3.64}$$

The formula for the estimated variance of the sample mean for stratified random sampling given in equation (3.62) is then given by:

$$v(\bar{y}_{st}) = \frac{\left(\sum_{h=1}^{L-1} W_h s_h\right)^2}{n'} - \frac{\sum_{h=1}^{L-1} W_h s_h^2}{N} \tag{3.65}$$

Similar derivations can be constructed for a greater number of take-all strata.

3.6 Applications

We can compare the optimal and proportional allocation algorithms in this chapter with results from simple random sampling by applying the algorithms to boundaries constructed on the populations given in section 2.6 of chapter 2. Unfortunately there is no information on the cost of sampling from each stratum for these populations, meaning that optimal allocation reduces to the Neyman allocation variant.

The results of this comparison are given in table 3.2, and support the derivations in section 3.3 that showed proportional allocation produces a lower variance of estimates that simple random sampling. Likewise the table shows that optimal (Neyman) allocation generally produces far lower variance estimates than proportional allocation or simple random sampling.

We can further investigate the relationship between the allocation strate-

Table 3.2: Variance and design effect of estimates using Optimal (Neyman) allocation, Proportional allocation, and Simple Random Sampling

	Variance of estimates			Design Effect	
Population	SRS	Prop	Optimal	Prop	Optimal
AAGIS					
- Farm Area (x)	129.0223	14.4684	0.6454	0.1121	0.0050
- Beef Cattle (y)	0.1047	0.0452	0.0108	0.4304	0.1033
SHS					
- Income (x)	0.9487	0.1308	0.0540	0.1378	0.0569
- Recreation (y)	0.0102	0.0076	0.0075	0.7509	0.7339
MU284					
- Real Estate (x)	7.25E+05	2.52E+05	1.22E+04	0.3431	0.0168
- Taxation (y)	1.15E+04	5.36E+03	256.2100	0.4615	0.0224
Debtors	9.38E+03	1.49E+03	154.0922	0.1583	0.0164
US Cities	7.9969	0.6469	0.2352	0.0809	0.0294
US Banks	906.7455	63.2557	29.2076	0.0697	0.0322
MRTS	2.0949	0.6508	0.0577	0.3102	0.0276
Simulated LN					
- Auxiliary (x)	0.0160	0.0020	0.0006	0.1245	0.0367
- Survey Var (y)	0.0168	0.0065	0.0035	0.3885	0.2085

gies in figure 3.1, by calculating the variance of estimates for differing correlations using the simulated bivariate Log-normal distribution discussed in chapter 2. This figure shows that proportional allocation always produces better results than simple random sampling. However optimal allocation does not necessarily produce better results when there is little correlation between the auxiliary information and survey population.

Optimal allocation uses the stratum variance (and cost of sampling units in a stratum) in order to determine the optimal sample size of each stratum. However when the correlation between the auxiliary information and the survey population is low, the stratum variance of the auxiliary information

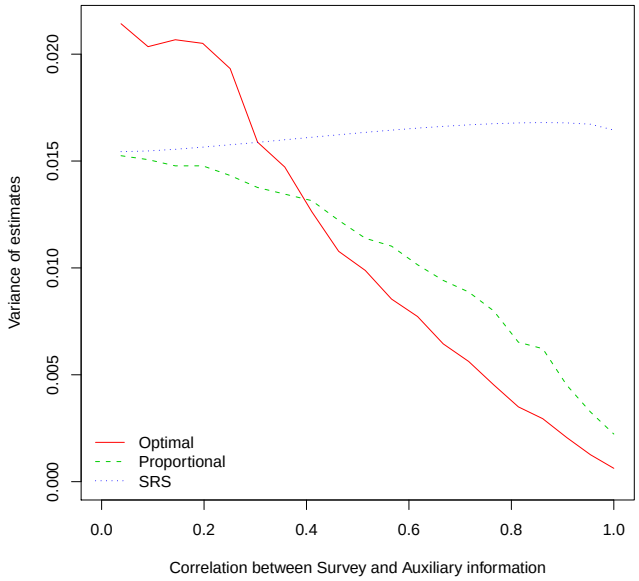

Figure 3.1: Variance of estimates using Optimal allocation, Proportional allocation, and Simple Random Sampling

may not provide a very good estimate of the corresponding variance of the survey population for that stratum. Optimal allocation consequently samples too few observations in strata with less variation in the auxiliary information, and too many observations in strata with greater variation in the auxiliary information.

The above situation has ultimately led to the result in figure 3.1, whereby optimal allocation produces poor result when there is a weak relationship between the auxiliary information and survey population, resulting in a variance of estimates that is worse than the variance from both proportional and simple random sampling. There are a number of model assisted approaches that could accommodate some of these issues, particularly for moderate correlation between the auxiliary information and survey variable. However many

of these are beyond the design-based approach of this thesis.

3.7 Summary

This chapter has covered several of the main issues concerning the allocation of sample units among strata for optimal stratification, and has given a comprehensive account for the derivation of the minimum variance and minimum cost estimators for optimal allocation. In particular the general formula for optimal allocation shows that we should take a larger sample n_h from a stratum when there is a larger number of population units N_h in a stratum, there is greater variation V' within a stratum, or the cost of sampling C' is cheaper for units within a stratum.

We have then applied the formulas for optimal allocation to stratified random sampling, and examined the relative merits of optimal allocation, proportional allocation, and simple random sampling. Section 3.4 also considered the relative loss (or increase) in variance of estimators from approximate values for the variance or sample size of strata that result in only approximately optimal allocation, and suggested that there may be instances where the simplicity of other sampling strategies may offset any estimated decrease in variance of the sample estimators. Finally we have derived formula for the construction of take-all strata, and have used these results as relevant sections of code in Appendix B.

The chapter does not provide a complete account of allocation issues in design based stratification, and the reader is directed to chapters in the broader based works of Cochran (1977), Lohr (1999), Thompson (1992),

Thompson (1997), and Särndal, Swensson & Wretman (1992), as well as the numerous journal articles on other related topics. However the chapter does provide a basis for work in subsequent chapters on the number and placement of boundary algorithms, and in particular the work of Dalenius (1950) in next chapter on the theory of optimal boundary placement.

Chapter 4

Stratum Boundaries

4.1 Overview

A solution for the placement of optimal stratum boundaries using Neyman allocation was first proposed in Dalenius (1950), and has been further clarified and generalised by a number of other authors (Dalenius & Hodges 1957, Cochran 1961, Horgan 2006). However the equations for the calculation of the optimal boundary points have been acknowledged as difficult to solve, and have consequently led to a number of approximations.

This chapter goes through the solution for the construction of optimal stratum boundaries, based upon the work of Dalenius (1950) and others. We then consider some of the issues with this solution, and apply the optimal boundary equations to a simple example of optimal stratification of a population using two strata.

4.2 Optimal Boundaries

One of the objectives of optimal stratification in addressing the problems considered in section 1.3 is to find the stratum boundaries k_0, ..., k_L that minimise equation (3.34) from section 3.2:

$$V(\bar{y}_{st}) = \frac{1}{n}\left(\sum_{h=1}^{L} W_h S_h\right)^2 - \frac{1}{N}\sum_{h=1}^{L} W_h S_h^2 \tag{4.1}$$

The outer boundaries k_0 and k_L can be set at the smallest y_1 and largest y_N values of the population respectively, meaning that we only need to find the internal stratum boundaries k_1, ..., k_{L-1}.

We can construct the formula for the conditional mean of stratum h as:

$$\begin{aligned}\bar{Y}_h &= \int_{k_{h-1}}^{k_h} y\left(\frac{f(y)}{\int_{k_{h-1}}^{k_h} f(s)ds}\right)dy \\ &= \frac{1}{W_h}\int_{k_{h-1}}^{k_h} yf(y)dy \end{aligned} \tag{4.2}$$

and the conditional variance as:

$$\begin{aligned}S_h^2 &= \int_{k_{h-1}}^{k_h} (y-\bar{Y}_h)^2\left(\frac{f(y)}{\int_{k_{h-1}}^{k_h} f(s)ds}\right)dy \\ &= \frac{1}{W_h}\int_{k_{h-1}}^{k_h} y^2 f(y)dy - \bar{Y}_h^2 \end{aligned} \tag{4.3}$$

where W_h is:

$$W_h = \int_{k_{h-1}}^{k_h} f(y)dy \tag{4.4}$$

The above equations (4.2), (4.3), and (4.4) clearly show that the conditional mean, conditional variance, and stratum weight for each stratum h are influenced only by the boundaries k_h and k_{h-1}. Likewise the boundary value k_h only appears in the above formulas for stratum h and stratum $h+1$.

The finite population correction is usually ignored in the derivation of the algorithms for the placement of optimal stratum boundaries, meaning it is sufficient to minimise the value of $W_h S_h$ in order to minimise $V(\bar{y}_{st})$ in equation (4.1). Therefore to find the minimum variance estimator, we only need to find the partial derivative of $W_h S_h$ with respect to k_h as follows:

$$\frac{\partial}{\partial k_h}\left(\sum_{h=1}^{L} W_h S_h\right) = \frac{\partial}{\partial k_h}\left(W_h S_h\right) + \frac{\partial}{\partial k_h}\left(W_{h+1} S_{h+1}\right) = 0 \tag{4.5}$$

We can use the result for the conditional variance of stratum h from equation (4.3) to construct the following:

$$W_h S_h^2 = \int_{k_{h-1}}^{k_h} y^2 f(y)dy - W_h \bar{Y}_h^2 \tag{4.6}$$

Substituting in equation (4.2) for the conditional mean and (4.4) for the stratum weight gives:

$$\begin{aligned} W_h S_h^2 &= \int_{k_{h-1}}^{k_h} y^2 f(y)dy - \frac{1}{W_h}\left(\int_{k_{h-1}}^{k_h} y f(y)dy\right)^2 \\ &= \int_{k_{h-1}}^{k_h} y^2 f(y)dy - \frac{\left(\int_{k_{h-1}}^{k_h} y f(y)dy\right)^2}{\int_{k_{h-1}}^{k_h} f(y)dy} \end{aligned} \tag{4.7}$$

We can then differentiate the above equation as follows:

$$\begin{aligned}\frac{\partial\left(W_h S_h^2\right)}{\partial k_h} &= S_h^2 \frac{\partial W_h}{\partial k_h} + 2W_h S_h \frac{\partial S_h}{\partial k_h} \\ &= S_h^2 f(k_h) + 2W_h S_h \left(\frac{k_h^2 f(k_h)}{f(k_h)} - \frac{(k_h f(k_h))^2}{f(k_h)^2} \right) \\ &= f(k_h)\left(k_h - \bar{Y}_h\right)^2 \end{aligned} \tag{4.8}$$

We then add $S_h^2 \partial W_h / \partial k_h$ to each side of the equation, and divide by $2S_h$, as follows:

$$\begin{aligned}\frac{1}{2S_h}\frac{\partial\left(W_h S_h^2\right)}{\partial k_h} + \frac{S_h}{2}\frac{\partial W_h}{\partial k_h} &= \frac{S_h}{2}\frac{\partial W_h}{\partial k_h} + W_h \frac{\partial S_h}{\partial k_h} + \frac{S_h}{2}\frac{\partial W_h}{\partial k_h} \\ &= S_h \frac{\partial W_h}{\partial k_h} + W_h \frac{\partial S_h}{\partial k_h} \\ &= \frac{\partial (W_h S_h)}{\partial k_h} \end{aligned} \tag{4.9}$$

and hence:

$$\begin{aligned}\frac{\partial (W_h S_h)}{\partial k_h} &= f(k_h)\frac{\left(k_h - \bar{Y}_h\right)^2}{2S_h} + \frac{S_h^2}{2S_h}\frac{\partial W_h}{\partial k_h} \\ &= \frac{1}{2} f(k_h)\frac{\left(k_h - \bar{Y}_h\right)^2}{S_h} + \frac{1}{2} f(k_h)\frac{S_h^2}{S_h} \\ &= \frac{1}{2} f(k_h)\frac{\left(k_h - \bar{Y}_h\right)^2 + S_h^2}{S_h} \end{aligned} \tag{4.10}$$

We similarly find:

$$\frac{\partial (W_{h+1} S_{h+1})}{\partial k_h} = -\frac{1}{2} f(k_h)\frac{(k_h - \bar{Y}_{h+1})^2 + S_{h+1}^2}{S_{h+1}} \tag{4.11}$$

We substitute the above equations (4.10) and (4.11) into equation (4.5), and

therefore find the minimum variance estimator occurs when:

$$\frac{S_h^2 + (k_h - \bar{Y}_h)^2}{S_h} = \frac{S_{h+1}^2 + (k_h - \bar{Y}_{h+1})^2}{S_{h+1}} \tag{4.12}$$

for $h = 1, \ldots, L-1$. However there are considerable difficulties in finding appropriate values of k_h to satisfy the above equations as the values of $\bar{Y}_h$ and S_h depend on k_h (as demonstrated in equations (4.2) and (4.3) at the start of this section). This has resulted in a number of approximations for the estimation of optimal stratum boundaries given in equation (4.12), and we will look at some of better known algorithms in subsequent chapters.

4.3 Illustration of Optimal Boundaries

The previous section derived the equations for the optimal boundary points, and noted that the equations proposed by Dalenius (1950) are difficult to solve. This is particularly the case as the number of strata increase, and hence increases the number of interacting values for the stratum means, stratum variance, and optimal stratum boundary points.

Optimal stratum boundaries can however be derived using the above equations for simple situations such as two strata, as there is only one boundary of interest and consequently far fewer dependencies among the variables. An example of this is given in table 4.1, which calculates the single optimal boundary point separating the two strata constructed on Household Income Before Taxes (in thousands of dollars) from the 2001 Survey of Household Spending (SHS).

Table 4.1: Calculation of optimal stratum boundary points for two strata on Household Income Before Taxes (thousands of dollars) from the 2001 Survey of Household Spending (SHS)

Boundary (k_h)	$\frac{S_h^2+(k_h-\bar{Y}_h)^2}{S_h}$	$\frac{S_{h+1}^2+(k_h-\bar{Y}_{h+1})^2}{S_{h+1}}$	$V(\bar{y}_{st})$
0	1594.57	6840.25	5487.84
500	843.50	6815.09	4589.11
1000	1084.91	7199.56	3502.74
1500	1757.63	7477.50	2834.33
2000	2588.79	7759.93	2382.51
2500	3522.75	8021.57	2086.54
3000	4514.51	8299.91	1887.26
3500	5553.08	8621.54	1751.87
4000	6670.91	8964.20	1665.42
4500	7917.32	9261.36	1620.26
5000	9239.28	9580.57	1600.59
5500	10625.09	9929.23	1602.11
6000	12172.32	10197.58	1620.07
6500	13705.65	10575.07	1653.09
7000	15438.13	10837.94	1692.90

The two sides of equation (4.12) listed in table 4.1 converge near a boundary value of $k_h = 5000$, and correspond to the minimum value of the variance in the table (using a sample size of 1606, being 10% of the population, and ignoring the finite population correction). We also plot the variance of the boundary points in figure 4.1, producing a smooth function that again has a minimum value of around $k_h = 5000$.

Unfortunately the equations for the optima boundary points can result in a number of challenges, such as local minima, as we increase the number of strata. We therefore need to consider other approximate solutions in order to estimate the optimal boundary points, and investigate some of the more

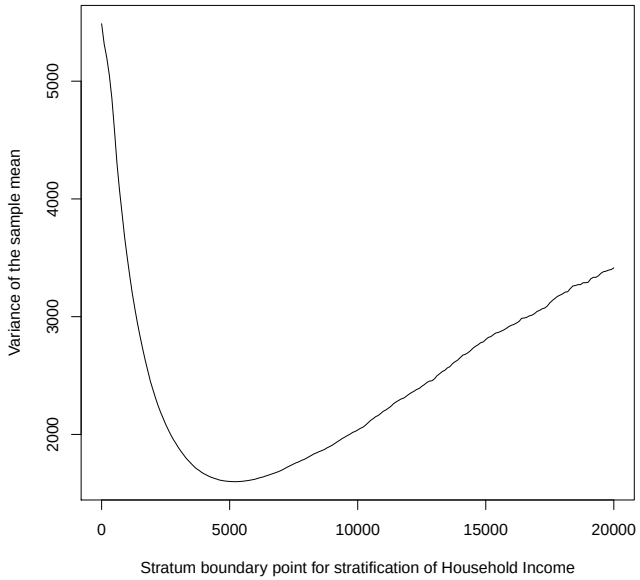

Figure 4.1: Variance of the sample mean for optimal stratification using two strata on Household Income Before Taxes (thousands of dollars) from the 2001 Survey of Household Spending (SHS)

popular approaches in the ensuing chapters.

4.4 Summary

This chapter has outlined the theory for the construction of optimal stratum boundaries, and briefly considered some of the difficulties in using these equations in order to find the optimal stratum boundary points. We have also illustrated the application of these equations to the placement of optimal stratum boundary points through a simple example of the stratification of the Household Income population into two strata.

This is one of the shortest chapters in this thesis, but its size belies its importance: the four subsequent chapters build on this chapter through

constructing algorithms to approximate the intractable solution proposed by Dalenius for the construction of optimal stratification boundaries. The three main approximations to the optimal boundary solution are covered in the next three chapters, being the cumulative square root of frequency, Ekman algorithm, and Lavallée-Hidiroglou approach. We then consider some other basic approaches in a fourth and final chapter on the placement of optimal stratum boundaries.

Chapter 5

Cumulative Square Root

5.1 Overview

The cumulative square root of frequency is a widely used algorithm for the construction of stratum boundaries, through creating boundaries using equal intervals on the cumulative square root of frequency scale. The algorithm was first proposed by (Dalenius & Hodges 1957) as a possible solution to the boundary problem, presented in chapter 4, and is considered to be a relatively straightforward approach to the estimation of optimal boundary points.

This chapter investigates the cumulative square root of frequency algorithm, and some of the assumptions that underpin this approach. In particular the algorithm makes a critical assumption that the distribution of values in each stratum is approximately uniform, and considers issues surrounding the construction of initial intervals for the calculation of the square root of frequencies.

We start in the next section with a quick review of the theory relating to this approach, and examine the practical implementation of the cumulative square root of frequency algorithm. The subsequent sections consider the issue of the initial intervals for the calculation of the square root of frequencies, and then several possible extensions to the approach.

5.2 Theory

The cumulative square root of frequency approach minimises the estimated overall variance of the sample mean (or sample total), as given in equation (4.1) of section 4.2, by incorporating several simplifying assumptions into the derivation of the optimal stratum boundaries. The first of these assumes that the distribution of values within a stratum is approximately uniform, and hence simplifies the estimated stratum variance to that of the uniform distribution:

$$S_h \approx \frac{1}{\sqrt{12}}(k_h - k_{h-1}) \tag{5.1}$$

We can also rewrite equation (4.4) for the stratum weight in section 4.2 using the first mean value theorem for integration as follows (Horgan 2006):

$$\begin{aligned} W_h &= \int_{k_{h-1}}^{k_h} f(y)dy \\ &= f(c)(k_h - k_{h-1}) \\ &= f_h(k_h - k_{h-1}) \end{aligned} \tag{5.2}$$

where $f_h = f(c)$ and c is on the interval $[k_h, k_{h+1}]$, and f_h must exist along the function $f(y)$ defined over the stratum interval. The resulting frequency f_h must also be greater than or equal to zero (as $f(y) \geq 0$), and, in a practical sense, the frequency f_h can be thought of as the frequency of the above assumed uniform distribution (although the uniform distribution assumption is not required for the relation to hold).

We established in section 4.2 that if the finite population correction is ignored, that it is sufficient to minimise the value of $W_h S_h$ in order to minimise the value of $V(\bar{y}_{st})$. The above results for the stratum variance and stratum weight can then be substituted into the sum of $W_h S_h$ as follows:

$$\begin{aligned} \sum_{h=1}^{L} W_h S_h &\approx \frac{1}{\sqrt{12}} \sum_{h=1}^{L} f_h (k_h - k_{h-1})^2 \\ &\approx \frac{1}{\sqrt{12}} \sum_{h=1}^{L} \left(\sqrt{f_h} (k_h - k_{h-1}) \right)^2 \end{aligned} \tag{5.3}$$

and therefore shows that the minimum is occurs when $\sqrt{f_h}(k_h - k_{h-1})$ is the same for all h:

$$\sqrt{f_1}(k_1 - k_0) = \sqrt{f_2}(k_2 - k_1) = \ldots = \sqrt{f_L}(k_L - k_{L-1}) \tag{5.4}$$

We also notice that:

$$\int_{k_{h-1}}^{k_h} \sqrt{f(y)} dy \approx \sqrt{f_h}(k_h - k_{h-1}) \tag{5.5}$$

and therefore the boundary points $k_0, \ldots, k_L$ are positioned so that:

$$\int_{k_0}^{k_1} \sqrt{f(y)}dy = \int_{k_1}^{k_2} \sqrt{f(y)}dy = \ldots = \int_{k_{L-1}}^{k_L} \sqrt{f(y)}dy \tag{5.6}$$

We now construct a function $G(y)$ such that:

$$G(y) = \int_{y_0}^{y} \sqrt{f(y)}dy \tag{5.7}$$

and hence:

$$\begin{aligned} G(k_h) - G(k_{h-1}) &= \int_{y_0}^{k_h} \sqrt{f(y)}dy - \int_{y_0}^{k_{h-1}} \sqrt{f(y)}dy \\ &= \int_{k_{h-1}}^{k_h} \sqrt{f(y)}dy \\ &\approx \sqrt{f_h}(k_h - k_{h-1}) \end{aligned} \tag{5.8}$$

If we set $H = G(k_L)$, being the integral of $\sqrt{f(y)}$ over the range $[k_0, k_L]$ (equivalent to the range $[y_0, y_N]$), then we find the cumulative square root of frequency:

$$H = \int_{k_0}^{k_L} \sqrt{f(y)}dy \tag{5.9}$$

and the square root of frequency for an individual stratum of:

$$G(k_h) - G(k_{h-1}) = \frac{H}{L} \tag{5.10}$$

Hence the approximately optimal boundary points occur at:

$$G(k_h) = \int_{k_0}^{k_h} \sqrt{f(y)}dy = \frac{hH}{L} \tag{5.11}$$

resulting in internal boundary values $k_1, \ldots, k_{L-1}$ on the cumulative square root of frequency scale of:

$$\frac{H}{L}, \frac{2H}{L}, \ldots, \frac{(L-1)H}{L} \tag{5.12}$$

This therefore demonstrates that an approximation to the optimal boundary points to minimise the variance of the estimator using the above assumptions can be found by taking equal intervals on the cumulative square root of frequency scale. The next two sections will go through the practical issues in the implementation of this algorithm, and in particular issues in the construction of initial intervals in order to calculate the cumulative square root of frequency.

5.3 Implementation

The cumulative square root of frequency algorithm is a relatively simple algorithm for the approximation of the optimal stratum boundary points. The implementation of the algorithm can be specified through a four stage process as follows:

Step 1: Ensure the population is sorted, and divide the range of the population into equal intervals

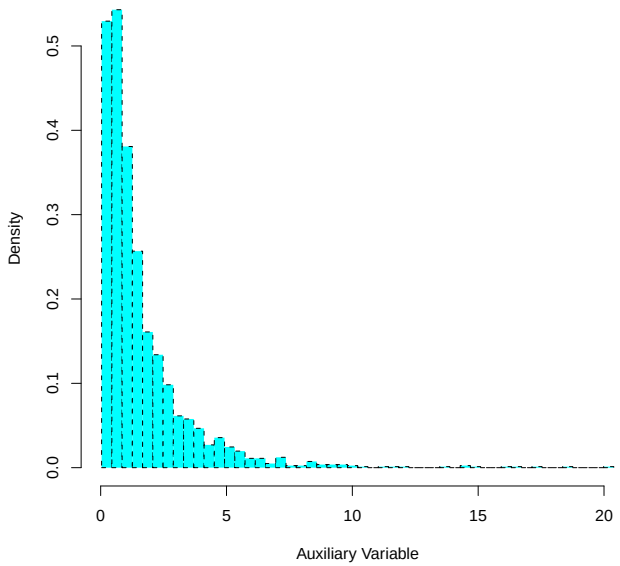

Figure 5.1: Histogram of initial intervals for a Simulated Bivariate Log-normal population ($N = 2000$)

Step 2: Calculate the frequency of values in each initial interval, as bounded by the above initial interval points

Step 3: Calculate the square root of the frequency $\sqrt{f}$ for each interval, and then construct these on a cumulative square of frequency scale

Step 4: Divide the cumulative square root of frequency scale into equal intervals, and then find the initial interval points closest to each of these divisions on this scale.

The above algorithm is implemented in the `csf` function in Appendix B, and is demonstrated in figures 5.1 to 5.3 using the simulated bivariate log-normal population. The histogram in figure 5.1 shows the construction of the initial interval points through creating fifty equal intervals along the

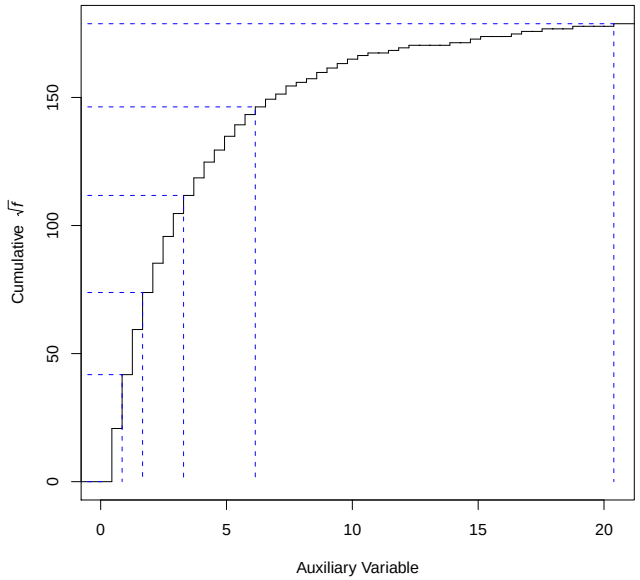

Figure 5.2: Construction of equal intervals on the cumulative square root of frequency scale for a Simulated Bivariate Log-normal population ($N = 2000$)

range of the population, and the frequency of values contained within each initial interval. The square root of these frequencies is then calculated, and constructed on the cumulative square root of frequency scale, as shown in figure 5.2. Equal divisions are then taken on this cumulative square root of frequency scale, and figure 5.3 shows the final boundaries overlaid on the initial intervals.

Unfortunately we are unlikely to find initial interval points that correspond exactly to the equal divisions on the cumulative square root of frequency scale. We therefore select the initial interval points which minimise the distance to the equal intervals on the cumulative square root of frequency scale.

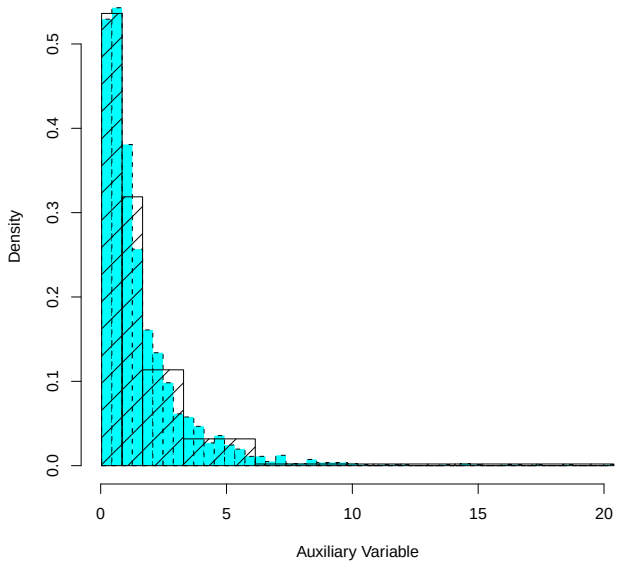

Figure 5.3: Histogram of initial intervals and resulting Cumulative Square Root of Frequency boundaries for a Simulated Bivariate Log-normal population ($N = 2000$)

We can also encounter issues when some of the initial intervals contain no actual values, and hence produce an initial interval with zero frequency. This creates duplicate values on the cumulative square root of frequency scale, albeit with differing values along the domain of the function. If one of these duplicate points is the closest point to one of the divisions on the cumulative square root of frequency scale, then we can select the first of the duplicates if the duplicate values are greater than the division, and likewise select the last if the duplicate values are less than the division on the cumulative square root of frequency scale.

Highly skewed populations can result in the same initial interval being the closest point to consecutive divisions on the cumulative square root of frequency, particularly when there are a smaller number of initial intervals.

This can be easily rectified through selecting another point; however this can produce rather unequal divisions on the cumulative square root of frequency scale. We consider these above issues further through investigating amendments to the cumulative square root of frequency rule in section 5.5 of this chapter.

Sometimes we may be provided with initial intervals of unequal sizes, or have a specific reason for constructing one or more initial intervals of a different size (such as for some administrative convenience). In order to account for this we first consider a base interval size u_1, which may simply be one of the initial intervals if all of the initial intervals are of different size. We then calculate the difference d_i between the base interval and the other intervals of different size as follows:

$$d_i = \frac{u_i}{u_1} \tag{5.13}$$

We can then correct for this difference by multiplying the value of the square root of frequency for these intervals by the square root of d_i when forming the cumulative square root of frequency scale.

Finally we have ignored the issue of the number of initial intervals to construct. Unfortunately there is no rule on the optimal number of initial intervals, and we consider this problem further in the next section.

5.4 Initial Intervals

The construction of initial intervals is a crucial stage in the implementation of the cumulative square root of frequency algorithm, as the final boundary points ultimately depend on these initial interval points. Cochran (1961) suggested constructing a large number in order to ensure there is a initial point close to the true optimal boundary, and Horgan (2006) discusses constructing sufficient boundaries such that there is some stability in the resulting boundary points. Hedlin (2000) however noted that there is still no rule or method for determining the optimal number of initial intervals and consequently initial interval points to construct.

In this section we investigate the effect of a change in the number of initial intervals on the resulting boundary points, through modelling the effects of changes in the number of initial intervals on the resulting boundary points. We then consider this in the context of the assumptions made as part of the cumulative square root of frequency rule, and attempt to add to the discussion on the appropriate number of initial intervals.

We first construct a graph of initial intervals and resulting boundary points in figure 5.4, through calculating the boundary points for between ten initial intervals and one thousand initial intervals on the bivariate log-normal distribution described in chapter 2. This shows the number of initial intervals has a marked effect on the placement of final boundaries for between ten and one hundred initial intervals, and then a lesser effect on the boundaries for between one hundred and one thousand initial intervals.

The diminishing marginal effect of changes in number of initial intervals

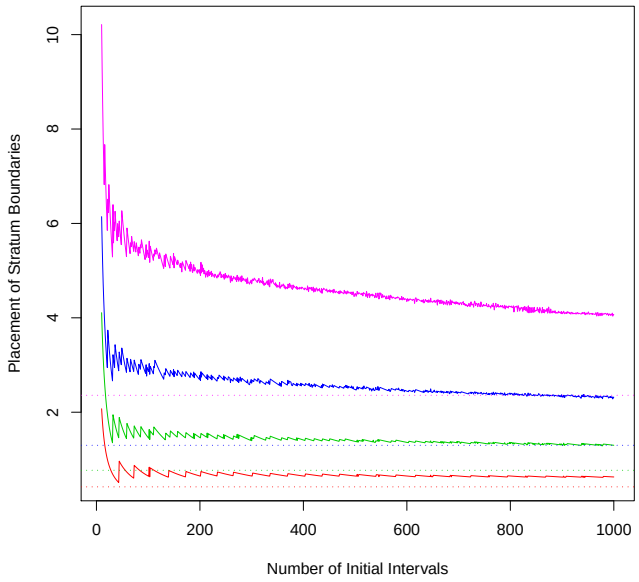

Figure 5.4: Cumulative Square Root of Frequency boundaries using different numbers of initial intervals on a Simulated Bivariate Log-normal population

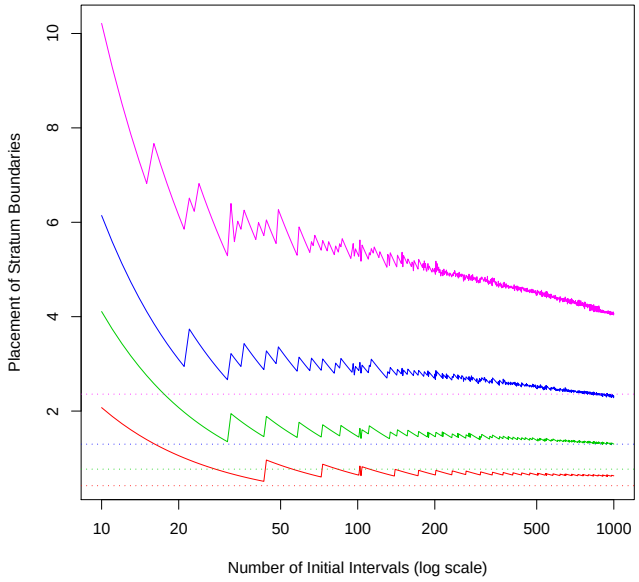

Figure 5.5: Cumulative Square Root of Frequency boundaries using different numbers of initial intervals on a Simulated Bivariate Log-normal population (displayed on a logarithmic scale)

on the stratum boundaries observed in figure 5.4 is characteristic of a log-linear relationship between two such variables. We therefore consider this in figure 5.5 through placing the auxiliary variable on a logarithmic scale, and find a close relationship between the movement of the final boundary points and the logarithm of the auxiliary variable. In particular this relationship is near linear when there are more than twenty initial intervals.

The above suggests that the stability in boundary points from selecting a higher number of initial intervals is somewhat artificial, and may instead be due to a log-linear relationship between the number of initial intervals and the final boundary points. Such a multiplicative relationship is not unexpected, as we need to increase from five hundred to one thousand initial intervals in order to perform the same type of split of initial intervals (into two) as would occur from a movement from fifty to one hundred initial intervals.

The boundaries also seem to eventually converge on positions corresponding to equal intervals on the cumulative frequency scale, as denoted by the four horizontal dotted lines on each figure. This is again not unexpected, as increases in the number of initial intervals should eventually result in intervals with only one value (bar identical values of the auxiliary variable). The square root of one is simply one, meaning the boundary points should gradually approach equal intervals on the cumulative frequency scale as the number of initial intervals increases.

We can examine the effect of this movement in boundary points in figure 5.6 by considering how increases in the number of initial intervals affect the variance of the estimates. This shows there are some substantial decreases in variance from selecting greater under of initial intervals up until fifty such

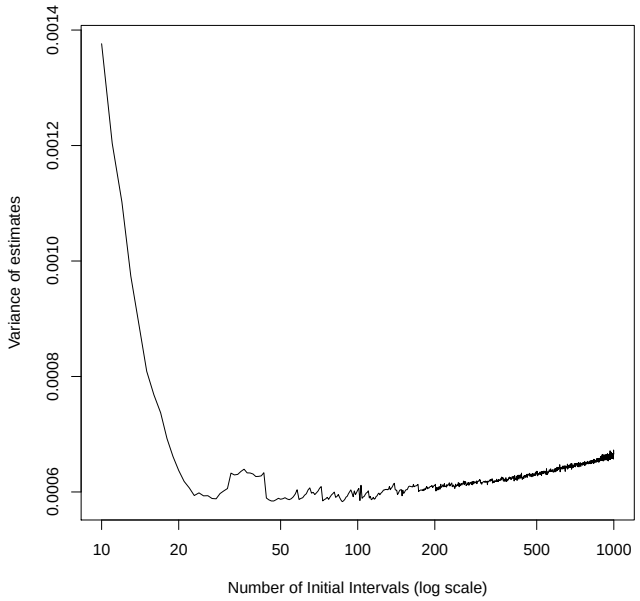

Figure 5.6: Variance of estimates for different numbers of initial intervals using the Cumulative Square Root of Frequency rule on a Simulated Bivariate Log-normal population (displayed on a logarithmic scale)

intervals, and then gradual increases in variance for more than fifty initial intervals. We also find through separating out the section for more than fifty initial intervals and through removing the logarithmic scale, that the increase in variance shown in figure 5.7 is near linear for between fifty initial intervals and one thousand initial intervals (and increases by some 13.3% over the range of the graph).

There is also considerable variation in stratum boundary points in figures 5.4 and 5.5, and we investigate this by taking the square of the diffence between successive estimates for the optimal boundary points and plotting this in figure 5.8. This shows that the variation in boundary points quickly decreases as the number of initial intervals increases, and consequently some of the "stability" in boundary points sought at the start of this section may

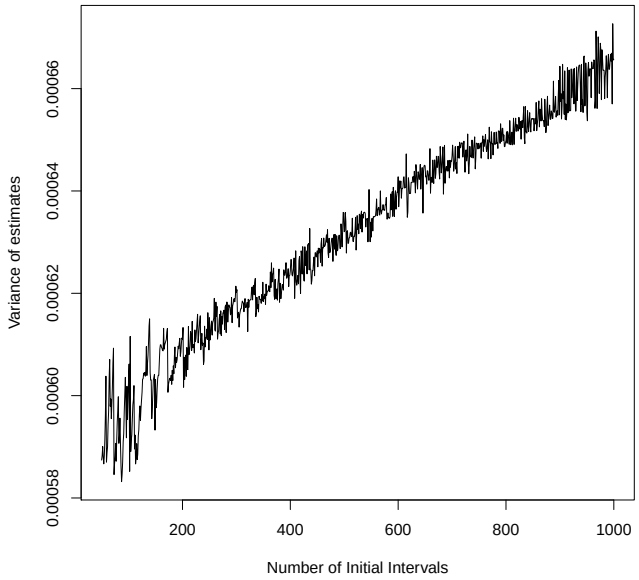

Figure 5.7: Variance of estimates for more than fifty initial intervals using the Cumulative Square Root of Frequency rule on a Simulated Bivariate Log-normal population

instead be due to the change in the boundary points from the change in the number of intervals and not from the variation between boundary estimates. The variation at the start of figure 5.8 possibly also contributes to some of the higher values for the variance of estimates in figure 5.6 for values of the number of initial intervals of less than twenty.

The above discussion of initial intervals suggests that a lower number of initial intervals should be used, perhaps around the value of fifty for this population (ten times the number of strata). We could also consider using some form of extrapolation between initial interval points in order to smooth some of the variation in initial intervals and resulting boundary points, and consider two options in the next two sections.

It is now useful to consider if there are any functional forms that may

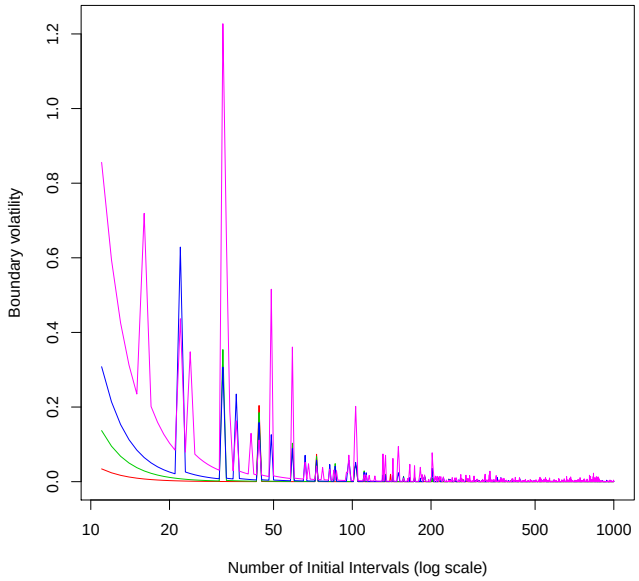

Figure 5.8: Volatility in boundaries for different numbers of initial intervals using the Cumulative Square Root of Frequency rule on a Simulated Bivariate Log-normal population (displayed on a logarithmic scale)

result in little or no change in stratum boundaries from a change in the number of initial intervals. If we denote the frequency of an interval i as f_i, then the weight of the interval relative to the sum of the square root of frequencies can be specified as:

$$\frac{\sqrt{f_i}}{\sum_{i=1}^{M} \sqrt{f_i}} \tag{5.14}$$

for M initial intervals. If we start with a simple situation of two strata ($h = 1, 2$) and two intervals, and split each interval such that $f_i = f_{i1} + f_{i2}$, then the following must hold in order for the relative weight of each original

interval to be preserved on the cumulative square root of frequency scale:

$$\frac{\sqrt{f_{11}}+\sqrt{f_{12}}}{\sqrt{f_{11}}+\sqrt{f_{12}}+\sqrt{f_{21}}+\sqrt{f_{22}}}=\frac{\sqrt{f_{11}+f_{12}}}{\sqrt{f_{11}+f_{12}}+\sqrt{f_{21}+f_{22}}} \tag{5.15}$$

We then multiple by each denominator:

$$\begin{aligned}&\left(\sqrt{f_{11}}+\sqrt{f_{12}}\right)\left(\sqrt{f_{11}+f_{12}}+\sqrt{f_{21}+f_{22}}\right)\\&\qquad=\left(\sqrt{f_{11}+f_{12}}\right)\left(\sqrt{f_{11}}+\sqrt{f_{12}}+\sqrt{f_{21}}+\sqrt{f_{22}}\right)\end{aligned} \tag{5.16}$$

and remove common values:

$$\left(\sqrt{f_{11}}+\sqrt{f_{12}}\right)\left(\sqrt{f_{21}+f_{22}}\right)=\left(\sqrt{f_{11}+f_{12}}\right)\left(\sqrt{f_{21}}+\sqrt{f_{22}}\right) \tag{5.17}$$

We square both sides:

$$\begin{aligned}&\left(f_{11}+2\sqrt{f_{11}f_{12}}+f_{12}\right)(f_{21}+f_{22})\\&\qquad=(f_{11}+f_{12})\left(f_{21}+2\sqrt{f_{21}f_{22}}+f_{22}\right)\end{aligned} \tag{5.18}$$

and again simplify to produce:

$$2\sqrt{f_{11}f_{12}}\,(f_{21}+f_{22})=2\sqrt{f_{21}f_{22}}\,(f_{11}+f_{12}) \tag{5.19}$$

Finally we rearrange the above to notice that:

$$\frac{\sqrt{f_{11}f_{12}}}{f_{11}+f_{12}}=\frac{\sqrt{f_{21}f_{22}}}{f_{21}+f_{22}} \tag{5.20}$$

must hold in order for any increase in initial intervals to produce the same relative weight of the original interval or stratum on the cumulative square root of frequency scale.

We can expand the above equation by recursively applying the result for greater numbers of initial intervals in order to produce the following general solution of:

$$\frac{\sqrt{f_{i1}f_{i2}}}{f_{i1}+f_{i2}} = Q \tag{5.21}$$

where Q is a constant. If we consider the situation where $f_{i2} = cf_{i1}$ for a constant c, then we can derive the following:

$$\begin{aligned} Q &= \frac{\sqrt{f_{i1}f_{i2}}}{f_{i1}+f_{i2}} \\ &= \frac{\sqrt{f_{i1}cf_{i1}}}{f_{i1}+cf_{i1}} \\ &= \frac{\sqrt{c}f_{i1}}{(1+c)f_{i1}} \\ &= \frac{\sqrt{c}}{(1+c)} \end{aligned} \tag{5.22}$$

and therefore shows that the equality holds. The only instances in which $f_{i2} = cf_{i1}$ is for the geometric and exponential memoryless distributions, and for the uniform distribution (when $c = 1$). This potentially means that the assumption of the uniform distribution within strata used in section 5.2 for the estimation of variance is also required in order to maintain the relative position of boundaries for different numbers of initial intervals.

5.5 Linear Interpolation

We noted in the previous section that there are two competing issues in the construction of optima boundary points using the cumulative square root of frequency rule:

- Using only a few initial intervals in order to avoid any movement in the boundary points, with the suggestion from the examples of around ten times the number strata.
- Creating enough initial intervals to ensure some stability and accuracy in the selection of boundary points

In this section we use linear interpolation in order to reduce some of the variation in the selection of stratum boundaries, by constructing a linear interpolant between successive initial interval points on the cumulative square root of frequency scale. This therefore allows us to select any point along the range, enabling us to used fewer initial intervals, and facilitating an exact match to the equal intervals on the cumulative square root of frequency scale.

We can construct a linear interpolant between successive points using the standard formula for linear interpolation as follows:

$$G(y) = \frac{G(y_i) - G(y_{i-1})}{y_i - y_{i-1}}(y - y_{i-1}) + y_{i-1} \tag{5.23}$$

where y represents values along the range of the population (and the initial interval boundary points y_i), and $G(y)$ is the corresponding value on the cumulative square root of frequency scale (as introduced in section 5.2). This

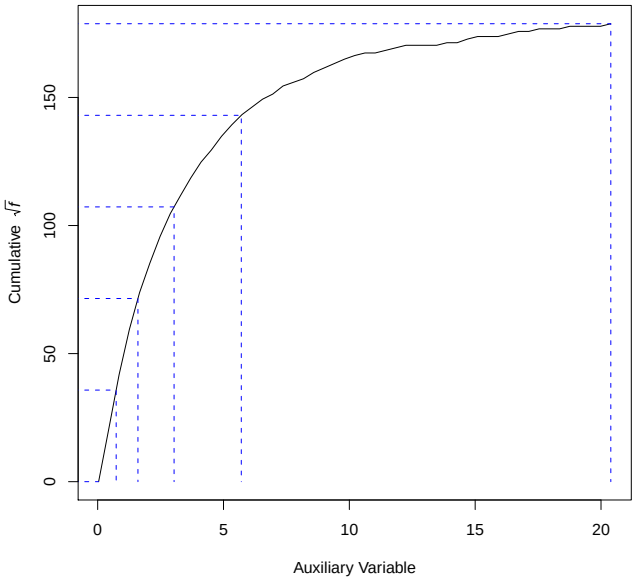

Figure 5.9: Construction of equal intervals using linear interpolation on the cumulative square root of frequency scale for a Simulated Bivariate Log-normal population ($N = 2000$)

can then be rearranged for the value of y as follows:

$$y = \frac{y_i - y_{i-1}}{G(y_i) - G(y_{i-1})}(G(y) - G(y_{i-1})) + G(y_{i-1}) \tag{5.24}$$

This is a more convenient form as we are attempting to find the values of y that relate to equal intervals on the cumulative square root of frequency scale, and hence the values of $G(y)$ are known.

We now modify the steps in section 5.3 to incorporate linear interpolation as follows:

Step 1: Ensure the population is sorted, and divide the range of the population into equal intervals

Step 2: Calculate the frequency of values in each initial interval, as

bounded by the above initial interval points

Step 3: Calculate the square root of the frequency $\sqrt{f}$ for each interval, and then construct these on a cumulative square of frequency scale

Step 4: Divide the cumulative square root of frequency scale into equal intervals, and then find the initial interval points that bound each of the equal intervals on the cumulative square root of frequency scale.

Step 5: Calculate the linear interpolant of the values of y for the given equal intervals using the initial intervals and corresponding values on the cumulative square root of frequency scale.

We demonstrate the implementation of this algorithm in figure 5.9, which shows the calculation of the exact points along the range corresponding to the equal intervals on the cumulative square root of frequency by interpolating values between the initial interval points. This algorithm is also simple to program, and is implemented as an option within the `csf` function contained in Appendix B.

5.6 Spline Interpolation

We can further attempt to improve on the estimated boundary points from the cumulative square root of frequency by using other forms of interpolation between the initial interval points. The linear interpolation algorithm given in the previous section is fast as easy; however we could instead generate a

smooth function of estimates across the initial interval points using the likes of a spline function.

We can use cubic interpolation to produce a smooth cumulative square root of frequency function; however in doing so we need to ensure that we a weakly increasing function across the range of the population. For this we utilise a monotonic cubic Hermite spline using the method of Fritsch & Carlson (1980), and implemented in the `splinefun` function of the `stats` package of the `R` programming language (we omit the actual derivation of this spline function as it is beyond the applied focus of this thesis).

We now incorporate the monotonic cubic Hermite spline into the algorithm given in previous section. Unfortunately we cannot easily reverse the function for the calculation of the monotone cubic spline in order to directly find the value of the stratum boundary for the given equal intervals on the cumulative square root of frequency scale, and instead need to use an iterative method as follows:

Step 1: Ensure the population is sorted, and divide the range of the population into equal intervals

Step 2: Calculate the frequency of values in each initial interval, as bounded by the above initial interval points

Step 3: Calculate the square root of the frequency $\sqrt{f}$ for each interval, and then construct these on a cumulative square of frequency scale

Step 4: Calculate the monotone cubic Hermite spline function using the

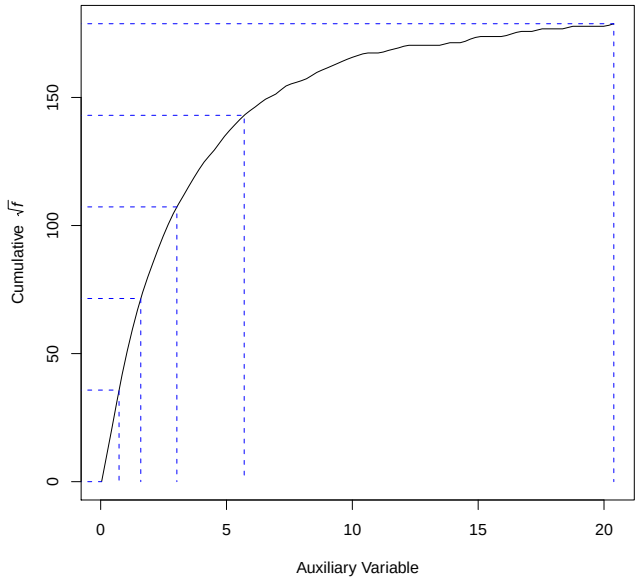

Figure 5.10: Construction of equal intervals using monotone cubic interpolation on the cumulative square root of frequency scale for a Simulated Bivariate Log-normal population ($N = 2000$)

initial interval points and the corresponding cumulative square root of frequency points.

Step 5: Divide the cumulative square root of frequency scale into equal intervals, and then find the initial interval points that bound each of the equal intervals on the cumulative square root of frequency scale.

Step 6: Search within these initial intervals, using the likes of the bisection method, to find the points along the range of the population that correspond to the equal intervals on the cumulative square root of frequency scale.

The implementation of the monotone spline algorithm is demonstrated

in figure 5.10, which shows the final cubic spline function and the points on the range that correspond to the spline function at the equal intervals on the cumulative square root of frequency scale. The monotone spline function is also implemented as an option within the `csf` function in Appendix B.

5.7 Results

We can compare the results from the cumulative square root of frequency algorithm with the linear and spline interpolation variants by deriving the estimated optimal stratum boundaries for the populations described in chapter 2. The results of this in table 5.1 show that the linear and spline extensions to the cumulative square root of frequency rule consistently produce better results than the original rule, with very little difference between the two interpolation variants.

These results are not unexpected: if the theory and assumptions underpinning the cumulative square root of frequency rule are reasonable, then any interpolation that results in boundaries that are a better match to the equal intervals on the cumulative square root of frequency scale should produce improvements in the variance of estimates. The original version of the cumulative square root of frequency rule did result in slightly better (lower variance) estimates for two of the survey populations; however this may have more to do with some of the allocation issues for optimal (or Neyman) allocation identified at the end of chapter 3.

We can also compare the performance of the various optimal boundary algorithms for different correlations between the survey and auxiliary infor-

Table 5.1: Design effect of estimates using the Cumulative Square Root of Frequency (CSF) rule, the Linear Interpolation extension to the CSF rule, and the Spline extension to the CSF rule

Population	CSF	Linear	Spline
AAGIS			
- Farm Area (x)	0.0050	0.0020	0.0020
- Beef Cattle (y)	0.1033	0.1198	0.1243
SHS			
- Income (x)	0.0569	0.0562	0.0561
- Recreation (y)	0.7339	0.7140	0.7068
MU284			
- Real Estate (x)	0.0168	0.0173	0.0174
- Taxation (y)	0.0224	0.0201	0.0201
Debtors	0.0164	0.0135	0.0132
US Cities	0.0294	0.0275	0.0275
US Banks	0.0322	0.0318	0.0318
MRTS	0.0276	0.0267	0.0265
Simulated LN			
- Auxiliary (x)	0.0367	0.0364	0.0365
- Survey Var (y)	0.2085	0.2086	0.2095

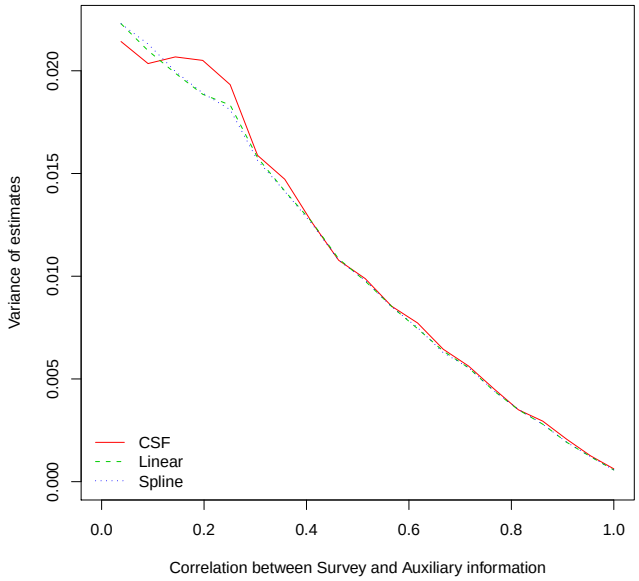

Figure 5.11: Variance of estimates using the Cumulative Square Root of Frequency (CSF) rule, the Linear Interpolation extension to the CSF rule, and the Spline extension to the CSF rule

mation in figure 5.11. This shows only slight differences between the three versions of the cumulative square root of frequency algorithm, with again the linear and spline variants outperforming the original cumulative square root of frequency rule.

The construction of a monotone cubic spline for the interpolation of points between initial intervals is however a slow and cumbersome procedure due to the recursive nature of the algorithm. This algorithm also produced very similar results to the linear interpolation algorithm and we therefore suggest that linear interpolation is possibly the superior of the two new variants of this algorithm considered in this chapter.

5.8 Summary

This chapter introduced one of the first major approximations to the intractable equations proposed by Dalenius (1950), the cumulative square root of frequency rule of Dalenius (1957). We have then considered the relative merits of this rule through first examining the theory that underlies this approach, and then looking at issues concerning the practical implementation of this algorithm.

We have derived the equations for the estimation of optimal boundaries using the cumulative square roof of frequency rule by combining the work of Dalenius (1957), Cochran (1977), and Horgan (2006), and demonstrated the implementation of this algorithm using the bivariate log-normal population from chapter 2. We have then examined the issue of the calculation of initial intervals in section 5.4, noting that there is no rule for the optimal number of initial intervals, and found that the construction of these initial intervals can have an effect on the final stratum boundary points.

We have found three main issues with the selection of initial intervals for the cumulative square root of frequency rule, being:

- Changes in boundaries from changes in the number of initial intervals: increasing the number of initial intervals can change in the resulting stratum boundaries (unless the underlying distribution is a geometric or exponential memoryless distribution, or the uniform distribution).

- Considerable boundary variation for smaller numbers of initial intervals: the selection of fewer initial intervals leads to fewer available boundary points, and hence somewhat unequal resulting intervals on

the cumulative square root of frequency scale.

- Cumulative frequency convergence: the square root of frequency scale could start to converge on the frequency scale for a sufficiently high number of initial intervals (particularly as this approaches infinity).

Our results from using various numbers of initial intervals tended to suggest that around fifty initial intervals would be suitable for our purposes, being ten times the desired number of final stratum boundaries. This therefore may provide a general guide to the number of initial intervals to use in similar highly skewed populations.

We also proposed two significant extensions to the cumulative square root of frequency rule, using linear interpolation and monotone cubic spline interpolation in order to select fewer initial intervals and reduce some of the boundary variation mentioned above. Both variants produced good results, with perhaps some preference for the linear variant due to the slow performance and iterative nature of the spline function.

We examine two major alternatives to the cumulative square root of frequency rule in the following chapters, being the Ekman approach and the Lavallée-Hidiroglou algorithm. We then compare the results in these sections with the results from this chapter.

Chapter 6

Ekman Algorithm

6.1 Overview

The previous chapter noted that the cumulative square root of frequency algorithm makes an important assumption concerning the distribution of values within a stratum, namely that they approximate a uniform distribution. We also considered issues around the construction of initial intervals for the calculation of the cumulative square root of frequency, but still observed that there is no general "rule" that gives the number of initial intervals.

An alternative approach the cumulative square root of frequency rule was proposed by Ekman (1959*b*), and further developed over a series of further papers by the same author (Ekman 1959*a*, 1963, 1969). This approach uses a Taylor expansion on the values given in equation (4.12):

$$\frac{S_h^2 + (k_h - \bar{Y}_h)^2}{S_h} = \frac{S_{h+1}^2 + (k_h - \bar{Y}_{h+1})^2}{S_{h+1}} \tag{6.1}$$

in order to derive the formula:

$$W_h(k_h - k_{h-1}) = C_L \tag{6.2}$$

for the approximate values of k_h for the optimal stratum boundary points. This differs from the derivation using the cumulative square root of frequency in that we are starting with the equation for the optimal stratum boundary points, instead of the equation for the variance of the overall sample estimator.

The Ekman approach has produced similar results in the past to the cumulative square root of frequency algorithm, and slightly superior results on some skewed populations (Hess, Sethi & Balakrishnan 1966). It has however also been noted that there are difficulties in finding an appropriate value of C_L that satisfies the Ekman approach (Cochran 1961).

This chapter will first go through the assumptions and derivation for the Ekman algorithm by combining and simplify the work of Ekman (1959*b*), and look at various issues relating to its implementation. We then construct an algorithm to iteratively solve for the estimated optimal stratum boundary points using the Ekman approach, and consider the extensions to this approach proposed by Hedlin (2000). Finally we suggest an alternative using kernel density functions, and then compare the three Ekman approaches with the results from the cumulative square root of frequency chapter.

6.2 Theory

The Ekman algorithm constructs equations to estimate the minimum variance stratum boundary points, given in equation (4.12) of section 4.2, by setting the value of $W_h(k_h - k_{h-1})$ to a constant C_L. In doing so, the derivation of the Ekman rule assumes that there exists a function $f(y)$ that is sufficiently differentiable to enable the construction of a Taylor expansion to derive the formula for the optimal stratum boundaries.

We begin the derivation of optimal stratum boundaries by first rearranging the formula for the stratum weight W_h from equation (4.4) in section 4.2:

$$W_h = \int_{k_{h-1}}^{k_h} f(y)dy \tag{6.3}$$

We can also rearrange the formula for the conditional mean in equation (4.2) to find $W_h\bar{Y}_h$ as follows:

$$W_h\bar{Y}_h = \int_{k_{h-1}}^{k_h} yf(y)dy \tag{6.4}$$

and likewise rearrange the results for the conditional variance in equation (4.3) to find $\bar{Y}_h^2$:

$$W_h(S_h^2 + \bar{Y}_h^2) = \int_{k_{h-1}}^{k_h} y^2 f(y)dy \tag{6.5}$$

We can now construct a function $I_i(k_{h-1}, k_h)$ to assist in deriving approximations to the equations for optimal stratum boundaries given in equation

(4.12) as follows (Ekman 1959a):

$$I_i(k_{h-1}, k_h) = \int_{k_{h-1}}^{k_h} (k_h - y)^i f(y) dy \tag{6.6}$$

We then solve this using equations (6.3), (6.4), and (6.5) above for $i = 0$:

$$\begin{aligned} I_0(k_{h-1}, k_h) &= \int_{k_{h-1}}^{k_h} f(y) dy \\ &= W_h \end{aligned} \tag{6.7}$$

for $i = 1$:

$$\begin{aligned} I_1(k_{h-1}, k_h) &= \int_{k_{h-1}}^{k_h} (k_h - y) f(y) dy \\ &= k_h \int_{k_{h-1}}^{k_h} f(y) dy - \int_{k_{h-1}}^{k_h} y f(y) dy \\ &= k_h W_h - W_h \bar{Y}_h \\ &= W_h \left(k_h - \bar{Y}_h \right) \end{aligned} \tag{6.8}$$

and for $i = 2$:

$$\begin{aligned} I_2(k_{h-1}, k_h) &= \int_{k_{h-1}}^{k_h} (k_h - y)^2 f(y) dy \\ &= k_h^2 \int_{k_{h-1}}^{k_h} f(y) dy - 2k_h \int_{k_{h-1}}^{k_h} y f(y) dy + \int_{k_{h-1}}^{k_h} y^2 f(y) dy \\ &= k_h^2 W_h - 2k_h W_h \bar{Y}_h + W_h (S_h^2 + \bar{Y}_h^2) \\ &= W_h \left(k_h^2 - 2k_h \bar{Y}_h + \bar{Y}_h^2 + S_h^2 \right) \\ &= W_h \left(S^2 + \left(k_h - \bar{Y}_h \right)^2 \right) \end{aligned} \tag{6.9}$$

We can use integration by parts on the function $I_i(k_{h-1}, k_h)$ in equation (6.6), using $k = k_h - k_{h-1}$ to simplify notation, and create a Taylor expansion for $i = 0$ as follows:

$$\begin{aligned} I_0(k_{h-1}, k_h) &= kf + \frac{k^2}{2!}f' + \frac{k^3}{3!}f'' + \frac{k^4}{4!}f''' + \frac{k^5}{5!}f^{(4)} + \dots \\ &= \sum_{t=1}^{\infty} \frac{k^t}{t!} f^{(t-1)} \end{aligned} \tag{6.10}$$

Likewise the Taylor expansion for $i = 1$ is:

$$\begin{aligned} I_1(k_{h-1}, k_h) &= \frac{k^2}{2!}f + \frac{k^3}{3!}f' + \frac{k^4}{4!}f'' + \frac{k^5}{5!}f''' + \frac{k^6}{6!}f^{(4)} + \dots \\ &= \sum_{t=2}^{\infty} \frac{k^t}{t!} f^{(t-2)} \end{aligned} \tag{6.11}$$

and for $i = 2$ is:

$$\begin{aligned} I_2(k_{h-1}, k_h) &= 2\left(\frac{k^3}{3!}f + \frac{k^4}{4!}f' + \frac{k^5}{5!}f'' + \frac{k^6}{6!}f''' + \frac{k^7}{7!}f^{(4)} + \dots\right) \\ &= 2\left(\sum_{t=3}^{\infty} \frac{k^t}{t!} f^{(t-3)}\right) \end{aligned} \tag{6.12}$$

We now substitute the results for W_h from equation (6.10) into equation (6.7) to find:

$$\begin{aligned} W_h = \Bigg((k_h - k_{h-1})f(k_{h-1}) &+ \frac{(k_h - k_{h-1})^2}{2!}f'(k_{h-1}) \\ &+ \frac{(k_h - k_{h-1})^3}{3!}f''(\xi_1)\Bigg) \end{aligned} \tag{6.13}$$

Similarly we substitute equation (6.11) into (6.8) for $W_h\left(k_h - \bar{Y}_h\right)$:

$$W_h\left(k_h - \bar{Y}_h\right) = \left(\frac{(k_h - k_{h-1})^2}{2!} f(k_{h-1}) + \frac{(k_h - k_{h-1})^3}{3!} f'(k_{h-1}) + \frac{(k_h - k_{h-1})^4}{4!} f''(\xi_2)\right) \tag{6.14}$$

and equation (6.12) into (6.9) for $W_h\left(S_h^2 + (k_h - \bar{Y}_h)^2\right)$:

$$W_h\left(S_h^2 + (k_h - \bar{Y}_h)^2\right) = 2\left(\frac{(k_h - k_{h-1})^3}{3!} f(k_{h-1}) + \frac{(k_h - k_{h-1})^4}{4!} f'(k_{h-1}) + \frac{(k_h - k_{h-1})^5}{5!} f''(\xi_3)\right) \tag{6.15}$$

where the ξ_i are points in the interval (k_{h-1}, k_h).

If we multiply equation (6.13) by (6.15) and subtract the square of (6.14), then the we obtain a value for the denominator in the optimal stratum boundary equation (4.12) in section 4.2 as follows (Ekman 1959*b*):

$$(W_h S_h)^2 = \frac{(k_h - k_{h-1})^4}{12}(f(k_{h-1})(f(k_{h-1}) + (k_h - k_{h-1}) f'(k_{h-1})) + R_1) \tag{6.16}$$

If then square the equation (6.15) in order to obtain a suitable corresponding numerator:

$$W_h^2\left(S_h^2 + (k_h - \bar{Y}_h)^2\right)^2 = \frac{(k_h - k_{h-1})^6}{9}\left(f(k_{h-1})\left(f(k_{h-1}) + \frac{f'(k_{h-1})}{2!}(k_h - k_{h-1})\right) + R_2\right) \tag{6.17}$$

The terms R_1 and R_2 are second order or higher in $(k_h - k_{h-1})$, and are able

to be ignored for large values of L (and hence small intervals $(k_h - k_{h-1})$). Finally we divide (6.17) by (6.16) to obtain:

$$\left(\frac{S_h^2 + (k_h - \bar{Y}_h)^2}{S_h}\right)^2 \sim \frac{4(k_h - k_{h-1})}{3} \cdot \frac{f(k_{h-1})(k_h - k_{h-1}) + \frac{f'(k_{h-1})}{2!}(k_h - k_{h-1})^2}{f(k_{h-1}) + f'(k_{h-1})(k_h - k_{h-1})} \quad (6.18)$$

The numerator of second factor on the right hand side has the first two terms of the Taylor expansion of W_h for the point k_{h-1}, and the denominator is the partial derivative of the numerator with respect to k_h. We therefore approximate the value of this expression using W_h and $f(k_h)$ as follows:

$$\left(\frac{S_h^2 + (k_h - \bar{Y}_h)^2}{S_h}\right)^2 \approx \frac{4(k_h - k_{h-1})W_h}{3f(k_h)} \quad (6.19)$$

We can also similarly derive the following:

$$\left(\frac{S_{h+1}^2 + (k_h - \bar{Y}_{h+1})^2}{S_{h+1}}\right)^2 \approx \frac{4(k_{h+1} - k_h)W_{h+1}}{3f(k_h)} \quad (6.20)$$

We now substitute the above equations (6.19) and (6.20) into the equation (4.12) for the placement of optimal stratum boundaries from section 4.2:

$$\frac{4(k_h - k_{h-1})W_h}{3f(k_h)} \approx \frac{4(k_{h+1} - k_h)W_{h+1}}{3f(k_h)} \quad (6.21)$$

which then gives:

$$W_h(k_h - k_{h-1}) \approx W_{h+1}(k_{h+1} - k_h) \quad (6.22)$$

Therefore the Ekman algorithm approximates the solution for the optimal (minimum variance) stratum boundaries by setting the boundary points k_h such that:

$$W_h(k_h - k_{h-1}) = C_L \tag{6.23}$$

where C_L is a constant that only depends on the number of strata L.

Unfortunately there is no rule that gives the value of C_L. Instead we go through an iterative process in the next section in order to find an appropriate value of C_L.

6.3 Implementation

We noted at the end of the previous section that there is no formula that gives the value of C_L for the Ekman algorithm, and hence we need to construct a process to find a series of boundary points such that the value of $W_{h+1}(k_{h+1} - k_h)$ is approximately constant. This section proposes an iterative algorithm to find the "best" value C_L for the given L number of strata by extending the work of Hedlin (2000) and Norland (1983), and derives some new results relating to the implementation of the Ekman algorithm.

We have already seen from equation (6.22) in the previous section that the Ekman algorithm attempts to find optimal boundary points such that:

$$W_h(k_h - k_{h-1}) \approx W_{h+1}(k_{h+1} - k_h) \tag{6.24}$$

and therefore:

$$W_1(k_1 - k_0) \approx W_L(k_L - k_{L-1}) \tag{6.25}$$

The population under consideration is usually discrete, and we are unlikely to find exact values of $W_h(k_{h+1} - k_h)$ to satisfy equation (6.23). Instead we construct an approximate value for each stratum h of

$$C_{L_h} = W_h(k_h - k_{h-1}) \tag{6.26}$$

where $C_{L_h} \approx C_L$. We note that by definition:

$$\sum_{h=1}^{L} W_h = \sum_{h=1}^{L} \frac{N_h}{N} = 1 \tag{6.27}$$

and also:

$$\sum_{h=1}^{L} k_h - k_{h-1} = k_L - k_0 \tag{6.28}$$

Therefore the smallest possible value of C_{L_h} is zero (when $k_h = k_{h-1}$ or $W_h = 0$), and the largest possible value of C_{L_h} is:

$$\begin{aligned} C_{L_h} &= \left(\sum_{h=1}^{L} W_h\right)\left(\sum_{h=1}^{L} k_h - k_{h-1}\right) \\ &= k_L - k_0 \end{aligned} \tag{6.29}$$

when $k_h = k_L$ and $k_{h-1} = k_0$ (where there is only one stratum).

We now use the above results to develop an iterative algorithm to find

the stratum boundaries k_h as follows:

Step 1: Select an initial value for C_L between zero and $k_L - k_0$.

Step 2: Find the value of k_1, and corresponding value of W_1, that produces the closest approximation of $C_{L_h} = W_h(k_h - k_{h-1})$ to the value of C_L.

Step 3: Repeat the previous step for all subsequent strata h up until stratum $h = L - 1$

Step 4: Calculate the value of $C_{L_L} = W_L(k_L - k_{L-1})$ for the final stratum using the boundary points k_{L-1} (calculated in the previous step) and k_L (the maximum value y_N)

Step 5: If the value of C_{L_L} is greater than C_L, then select a larger value for C_L and return to step 2. Likewise if the value of C_{L_L} is less than C_L, then select a smaller value for C_L and return to step 2. Otherwise if the value of C_{L_L} is the closest approximation to the value of C_L then the algorithm ends.

We are unlikely to find a value of C_L that results in the same values of $C_{L_h} = W_h(k_h - k_{h-1})$ for all strata, and hence need to settle on the set of boundary points that gives the closest approximation to this equality.

We implement the above algorithm in the `ekman` function in Appendix B by retaining upper and lower bounds on the value of C_L, and use a straightforward bisection method to find new values of C_L. There are more efficient search methods than the bisection method, but this is a separate issue to that of the Ekman algorithm and the bisection method will be sufficient for

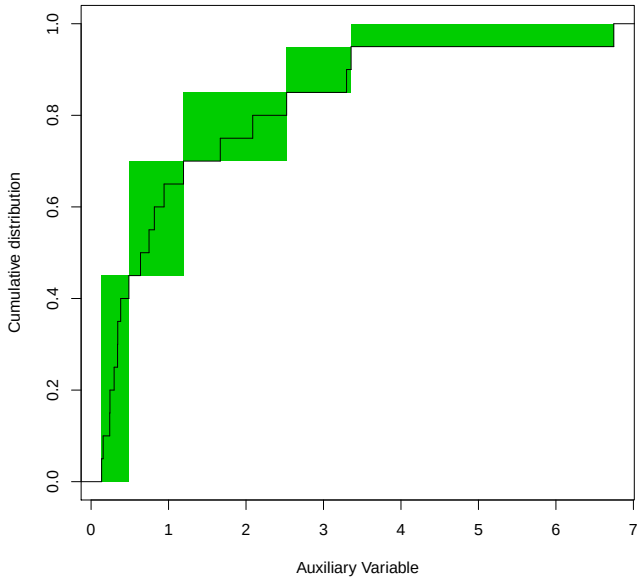

Figure 6.1: Construction of stratum boundaries using the Ekman algorithm for a Simulated Bivariate Log-normal population ($N = 20$)

our purposes. The algorithm then stops when the upper and lower bounds converge on the same point, (within a certain level of tolerance).

We also demonstrate the implementation of the Ekman algorithm in figure 6.1 on a simulated bivariate log-normal population with a population size of 20. The selection of such as small population size is deliberate, as it will be used throughout this chapter to help illustrate some of the differences between the various Ekman based algorithms.

The value of $W_h(k_h - k_{h-1})$ can be visually represented on the cumulative distribution in figure 6.1 as the area of the rectangle between the upper and lower boundaries for the stratum boundaries and respective cumulative frequency points. The iterative algorithm is then searching for a value of this area, referred to as "Ekman rectangles" by Hedlin (2000), so that all

rectangles presented in figure 6.1 are approximately of the same size.

The above iterative algorithm will always converge on one global minimum, as the value of C_L is weakly monotonic (increasing) function in k_h. If we increase the value of k_h to $k'_h = k_h + \epsilon$ then:

$$\begin{aligned} C'_{L_h} &= W_h(k_h + \epsilon - k_{h-1}) \\ &= W_h(k_h - k_{h-1}) + W_h\epsilon \\ &= C_{L_h} + W_h\epsilon \\ &\geq C_{L_h} \end{aligned} \tag{6.30}$$

for $\epsilon \geq 0$. We likewise find for subsequent strata that:

$$\begin{aligned} C'_{L_{h+1}} &= W_{h+1}(k_{h+1} - k_h - \epsilon) \\ &= W_{h+1}(k_h - k_{h-1}) - W_{h+1}\epsilon \\ &= C_{L_{L+1}} - W_{h+1} \\ &\leq C_{L_{h+1}} \end{aligned} \tag{6.31}$$

This shows that an increase in k_h cannot result in a lower value of $C_{L_{h+1}}$, and therefore cannot result in lower subsequent stratum boundaries. Hence any increase in the first stratum boundary must result in subsequent stratum boundaries that are greater or equal to the existing stratum boundaries.

One item thus far overlooked is the initial value of C_L to use in the above method. A possible approach is to first divide each side of equation (6.23)

by $(k_L - k_0)$ and then sum over L as follows (Ekman 1959*b*):

$$\sum_{h=1}^{L} W_h \frac{(k_h - k_{h-1})}{k_L - k_0} = \frac{LC_L}{k_L - k_0} \tag{6.32}$$

We can estimate the range of the stratum values relative to the overall range as:

$$\frac{k_h - k_{h-1}}{k_L - k_0} \approx \frac{1}{L} \tag{6.33}$$

Substituting equations (6.33) and (6.27) into (6.32) results in:

$$\frac{1}{L} \approx \frac{LC_L}{(k_L - k_0)} \tag{6.34}$$

which can be rewritten as:

$$C_L \approx \frac{(k_L - k_0)}{L^2} \tag{6.35}$$

Therefore we can use this value as an initial approximation to the value of C_L in our iterative method above, improving the speed of the iterative algorithm.

6.4 Extended Approach

One of the interesting results from the Ekman algorithm presented in the previous section is that each approximation to C_L of $C_{L_h} = W_h(k_h - k_{h-1})$ can be represented as an area on the cumulative frequency $F(y)$ for the

respective population as follows:

$$F(y) = \int_{y_0}^{y} fydy \tag{6.36}$$

where:

$$\begin{aligned} W_h &= \int_{k_0}^{k_h} f(y)dy - \int_{h_0}^{h_{h-1}} f(y)dy \\ &= F(k_h) - F(k_{h-1}) \end{aligned} \tag{6.37}$$

as demonstrated in figure 6.1 through the construction of stratum boundaries for a simulated bivariate log-normal population of size $N = 20$.

We now extend the Ekman algorithm by constructing the cumulative distribution function of a discrete population as a piecewise continuous step function with appropriate vertical and horizontal projections from the population values (Hedlin 2000). To do this we first let k_h equal any point along the interval $[k_0, k_L]$, and then allow the cumulative frequency $F'(k_h)$ to equal any corresponding point over the interval $[0, 1]$ on the piecewise continuous step function such that:

$$\sum_{i=1}^{h} W_i \leq F'(k_h) \leq \sum_{i=1}^{h} W_{i+1} \tag{6.38}$$

The value of C_L now becomes:

$$C_L = \left(F'(k_h) - F'(k_{h-1})\right)(k_h - k_{h-1}) \tag{6.39}$$

We note that this still holds for the procedure detailed in the previous section

for $W_h = F(k_h) - F(k_{h-1}) = F'(k_h) - F'(k_{h-1})$. This allows us to select any points along this function in order to find the constant value of C_L such that $C_L = C_{L_h}$ for all $h = 1, \ldots, L$, and will result in a pair of values for $F'(k_h)$ and k_h along one of the adjoining horizontal or vertical lines.

We can then apply an amended iterative procedure, based on the procedure given in the previous section:

Step 1: Select an initial value for C_L between zero and $k_L - k_0$ (as before).

Step 2: Find the values of k_1 and $F'(k_1)$ along the piecewise step function that result in $C_L = (F'(k_h) - F'(k_{h-1}))(k_h - k_{h-1})$.

Step 3: Repeat the previous step for all subsequent strata h up until stratum $h = L - 1$, bearing in mind that $F'(k_{h-1})$ and k_{h-1} are now points along the continuous step function (and not restricted to actual population values)

Step 4: Calculate the value of $C_{L_L} = (F'(L_h) - F'(k_{L-1}))(k_L - k_{L-1})$ for the final stratum using the points k_{L-1} and $F(k_{L-1})$ (calculated in the previous step) and the points $F'(L_h) = F(L_h) = 1$ and k_L (the maximum values for y_N)

Step 5: If the value of C_{L_L} is greater than C_L, then select a larger value for C_L and return to step 2. Likewise if the value of C_{L_L} is less than C_L, then select a smaller value for C_L and return to step 2. Otherwise if $C_L = C_{L_L}$, within an acceptable level of tolerance, then the algorithm ends.

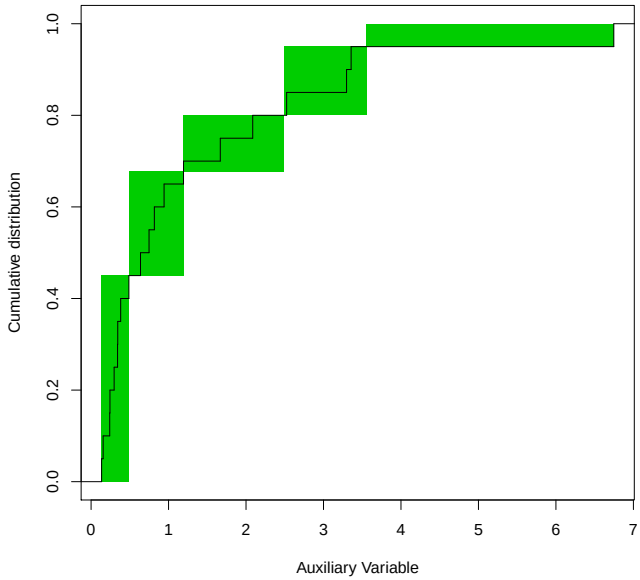

Figure 6.2: Construction of stratum boundaries using the extended Ekman algorithm for a Simulated Bivariate Log-normal population ($N = 20$)

The above represents a generalisation of the procedure used in the previous section, and will result in Ekman algorithm based approximations for the optimal boundary points.

We implement of the extended Ekman algorithm in figure 6.1 on a simulated bivariate log-normal population with a population size of 20. The small population size enables us to clearly see the selection of points on horizontal and vertical lines for the third and fifth stratum boundaries, and facilitates the calculation of a single constant value C_L for the area of the "Ekman rectangles" discussed in the previous section.

Given this continuous function, we can find some interesting limits on the value of $(F'(k_h) - F'(k_{k-1}))(k_h - k_{h-1})$. We find the maximum boundary points for the first stratum by noticing the total "area" over the cumulative

distribution function is:

$$LC_L = \sum_{h=1}^{L} W'_h K_h \tag{6.40}$$

where $K_h = k_h - k_{h-1}$ and $W'_h = F'(k_h) - F'(k_{k-1})$ (with both values defined over the continuous function). We further define K such that:

$$\begin{aligned} K &= \sum_{h=1}^{L} K_h \\ &= \sum_{h=1}^{L} k_h - k_{h-1} \\ &= k_L - k_0 \end{aligned} \tag{6.41}$$

We can now use a Lagrange multiplier, as introduced in section 3.2, to find the maximum value of C_L under the Ekman rule as follows:

$$\begin{aligned} \Lambda(W'_h, K_h, \lambda_1, \lambda_2) = \sum_{i=1}^{L} W'_i K_i &- \lambda_1 \left(\sum_{i=1}^{L} W'_i - 1 \right) \\ &- \lambda_1 \left(\sum_{i=1}^{L} K_i - K \right) \end{aligned} \tag{6.42}$$

The partial derivates with respect to W'_h, K_h, λ_1, and λ_2 are as follows:

$$\frac{\partial \Lambda}{W'_h} = W'_h - \lambda_1 = 0 \tag{6.43}$$

$$\frac{\partial \Lambda}{K_h} = K_h - \lambda_2 = 0 \tag{6.44}$$

$$\frac{\partial \Lambda}{\lambda_1} = \sum_{h=1}^{L} W'_h - 1 = 0 \tag{6.45}$$

$$\frac{\partial \Lambda}{\lambda_2} = \sum_{h=1}^{L} K_h - K = 0 \tag{6.46}$$

We find by substituting equation (6.43) into (6.45) and equation (6.44) into (6.46), and dividing both results by L, that:

$$\lambda_1 = \frac{1}{L} \tag{6.47}$$

$$\lambda_2 = \frac{K}{L} \tag{6.48}$$

Substituting the value of λ_1 from equation (6.47) back into (6.43) gives

$$W'_h = \frac{1}{L} \tag{6.49}$$

Likewise substituting the value of λ_2 from equation (6.48) back into equation (6.44) gives:

$$\begin{aligned} K_h &= \frac{K}{L} \\ &= \frac{(k_L - k_0)}{L} \end{aligned} \tag{6.50}$$

Therefore the maximum value of C_L occurs at:

$$\begin{aligned} C_L &= \frac{1}{L}\sum_{h=1}^{L} W'_h K_h \\ &= \frac{1}{L}\sum_{h=1}^{L} \frac{1}{L}\frac{(k_L - k_0)}{L} \\ &= \frac{1}{L}\frac{(k_L - k_0)}{L} \end{aligned} \tag{6.51}$$

This is the same value as the first approximation to the value of C_L given in equation (6.35) at the end of the last section (albeit now for a continuous function). Furthermore we observe that these maximum values for the first stratum occur on the diagonal between $(k_0, 0)$ and $(k_L, 1)$, and hence suggest that the maximum values of C_L would only be achieved for a population matching that of a uniform distribution.

In the next section we go through another possible extension to the Ekman algorithm using kernel density estimators. We then compare all three algorithms in section 6.6.

6.5 Kernel Density Approach

We can further develop the "extended" Ekman algorithm detailed in the previous section, by proposing a new method that approximates the cumulative distribution function for a population using a kernel density estimator. We can then apply the extended Ekman algorithm in the same manner as the previous section in order to find the Ekman approximations to the optimal boundary points.

We can construct a cumulative distribution for a population of size N using a kernel estimator as follows (Wand & Jones 1995):

$$F_{ker}(x) = \frac{1}{N}\sum_{i=1}^{N} P\left(\frac{x-y_i}{b}\right) \tag{6.52}$$

where $P(t) = \Phi(t)$, the cumulative distribution function of the normal distribution, and b is the bandwidth of the kernel estimator.

The cumulative distribution function for the Ekman algorithm is only defined between the points y_0 and y_N, and therefore we need to reweight the kernel $P(t)$ to points between y_0 and y_N as follows:

$$P'\left(\frac{x-y_i}{b}\right) = \frac{P\left(\frac{x-y_i}{b}\right) - P\left(\frac{y_0-y_i}{b}\right)}{P\left(\frac{y_N-y_i}{b}\right) - P\left(\frac{y_0-y_i}{b}\right)} \tag{6.53}$$

We then define the cumulative distribution using the kernel estimator as:

$$F^*_{ker}(x) = \begin{cases} 0 & \text{if } x \leq y_0 \\ P'\left(\frac{x-y_i}{b}\right) & \text{if } y_0 < x \leq y_N \\ 1 & \text{if } x > y_N \end{cases} \tag{6.54}$$

The bandwidth b of the kernel estimator can be estimated using the normal reference rule (Martinez & Martinez 2002):

$$b = \left(\frac{4}{3}\right)^2 SN^{-0.2} \tag{6.55}$$

which can be estimated using the interquartile range ($\hat{S} = \text{IQR}/1.348$) as

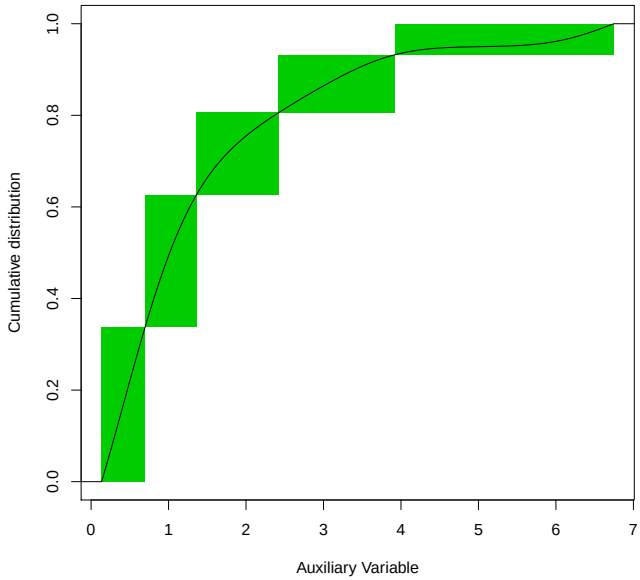

Figure 6.3: Construction of stratum boundaries using the Kernel Density Ekman algorithm for a Simulated Bivariate Log-normal population ($N = 20$)

follows:

$$\hat{b} = 0.768 \times \text{IQR} \times N^{-0.2} \tag{6.56}$$

Silverman (1986) recommends using the smaller of the two estimates for the bandwidth in the kernel estimator.

The above results then allow us to calculate the boundary values for the Ekman algorithm using the general algorithm given in the previous section. We demonstrate this in figure 6.3 through the construction of a smooth kernel density function, and then use this to find a constant value of C_L, and hence construct the stratum boundary points. We also implement the bounded version of the cumulative distribution function using a kernel estimator in the `ekman.kernel` function in Appendix B, and compare the performance of

the standard Ekman algorithm, extended Ekman algorithm, and the kernel based Ekman algorithm in the next section.

6.6 Results

We can implement the Ekman algorithm, the extended Ekman algorithm, and the kernel density based algorithm on the populations in chapter 2, and summarise the results of this comparison in table 6.1. This shows very similar results for all variants of the Ekman algorithms, and considerably similarities with the results for the cumulative square root of frequency algorithm. The Ekman algorithm performs slightly better on the AAGIS farm area and US banks populations, and the cumulative square root of frequency performs slightly better on the SHS household income population. However the reverse occurs for the relevant survey populations, with the Ekman producing better results on the SHS recreation expenditure population, and the cumulative square root of frequency rule performing better on the AAGIS beef cattle population.

We can also compare the three Ekman algorithms with the cumulative square root of frequency algorithms by considering the variance of estimates for different correlations between the auxiliary information and survey population. Figure 6.4 again shows near identical results for all three Ekman algorithms, and some overall improvements on the values for the cumulative square root of frequency algorithms. This is particularly case for lower correlations between the auxiliary information and survey population, with some difference between the performance of the Ekman algorithms and the

Table 6.1: Design effect of estimates using the Ekman algorithm, the Extended Ekman algorithm, the Kernel Density Ekman algorithm, and the Cumulative Square Root of Frequency algorithms

Population	Ekman	Extended	Kernel	CSF	Linear
AAGIS					
- Farm Area (x)	0.0015	0.0015	0.0015	0.0050	0.0020
- Beef Cattle (y)	0.1557	0.1557	0.1557	0.1033	0.1198
SHS					
- Income (x)	0.0595	0.0589	0.0588	0.0569	0.0562
- Recreation (y)	0.6798	0.6770	0.6811	0.7339	0.7140
MU284					
- Real Estate (x)	0.0179	0.0179	0.0176	0.0168	0.0173
- Taxation (y)	0.0174	0.0184	0.0185	0.0224	0.0201
Debtors	0.0147	0.0147	0.0147	0.0164	0.0135
US Cities	0.0265	0.0265	0.0263	0.0294	0.0275
US Banks	0.0308	0.0308	0.0312	0.0322	0.0318
MRTS	0.0323	0.0324	0.0324	0.0276	0.0267
Simulated LN					
- Auxiliary (x)	0.0371	0.0371	0.0371	0.0367	0.0364
- Survey Var (y)	0.2027	0.2031	0.2037	0.2085	0.2086

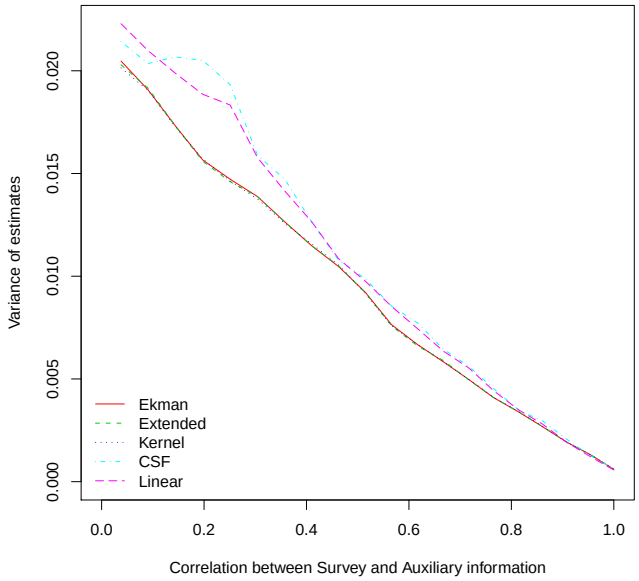

Figure 6.4: Variance of estimates using the Ekman algorithm, the Extended Ekman algorithm, the Kernel Density Ekman algorithm, and the Cumulative Square Root of Frequency algorithms

cumulative square root of frequency results.

The extended Ekman algorithm presented in this chapter is only marginally more complicated than the original Ekman algorithm, and may provide some benefits in producing an exact solution for smaller populations through decreasing the reliance on observed data points. The kernel density version also reduces this reliance, however is considerably more complicated to implement. We therefore we may prefer use the extended version of this algorithm in such instances.

6.7 Summary

This chapter has considered the Ekman approach for the construction of optimal stratum boundaries through deriving the equations for the estimation of these boundaries and then proposing a number of iterative algorithms to solve these equations. Several authors have experienced difficulties arriving at a solution to these equations; however we show through some basic derivations that the constant value C_L is a weakly increasing function in k_h, and consequently our algorithm will converge on a single solution.

We have examined three variants of the Ekman algorithm: the original algorithm of Ekman (1959*b*), the extended approach of Hedlin (2000), and a new kernel density based algorithm. All three algorithms tend to produce slightly better results than the cumulative square root of frequency algorithms of the previous chapter, and we observe very similar results among the three algorithms for sufficiently large populations

The kernel density algorithm is considerably more complicated than the other two Ekman algorithms presented in this chapter, but may have the potential for further development through further examining the effect of the kernel bandwidth on the resulting stratum boundary points. However we conclude that the ease and speed of implementation of the original and extended versions may make these preferable to the current version of the kernel based approach.

Chapter 7

Lavallée-Hidiroglou Algorithm

7.1 Overview

Lavallée & Hidiroglou (1988) proposed an iterative procedure to find optimal stratum boundaries using a combination of procedures from Sethi (1963) and Hidiroglou (1986). The approach is particularly suited to the stratification of highly skewed populations, such as those encountered in business and agricultural surveys, and is generally used to determine stratum boundaries by minimising the sample size for a required coefficient of variation of the estimator.

The Lavallée-Hidiroglou algorithm solves for the optimal stratum boundaries by taking the partial derivatives of the variance from stratified random sampling with respect to each boundary k_h by restating the equation for the variance of the estimates as follows:

$$n = NW_L + \frac{\sum_{h=1}^{L-1} W_h^2 S_h^2 / a_h}{\bar{Y}^2 c^2 + \sum_{h=1}^{L-1} W_h S_h^2 / N} \tag{7.1}$$

However Detlefsen & Veum (1991) experienced a number of issues in the application of the Lavallée-Hidiroglou algorithm using Neyman allocation, and in particular found the algorithm was slow to converge or did not converge on a solution to the equations. Consequently Kozak (2004) suggested an alternative implementation of the Lavallée-Hidiroglou algorithm, based on the work of Lednicki & Wieczorkowski (2003).

This chapter will consider the derivation of the Lavallée-Hidiroglou algorithm, and examine the implementation of this algorithm. We will then compare the results for this algorithm in the previous two chapters, and consider the relative merits of the approach.

7.2 Theory

The Lavallée-Hidiroglou approach proposes a derivation for optimal stratum boundaries using the formula for the variance of the population mean given in equation (3.26) of chapter 3 as follows:

$$V(\bar{y}_{st}) = \sum_{h=1}^{L} \frac{W_h^2 S_h^2}{n_h} - \sum_{h=1}^{L} \frac{W_h S_h^2}{N} \tag{7.2}$$

We accommodate the possibility of a single "take-all" stratum L by setting the sample size as:

$$n_h = (n - N_L)\, a_h \tag{7.3}$$

where a_h is the allocation rule used and N_L is a take all stratum (as discussed in chapter 3). The equation for the variance then becomes:

$$\begin{aligned} V(\bar{y}_{st}) &= \sum_{h=1}^{L} \frac{W_h^2 S_h^2}{(n - N_L)a_h} - \sum_{h=1}^{L} \frac{W_h S_h^2}{N} \\ &= \frac{1}{(n - N_L)} \sum_{h=1}^{L} \frac{W_h^2 S_h^2}{a_h} - \sum_{h=1}^{L} \frac{W_h S_h^2}{N} \end{aligned} \tag{7.4}$$

and we then rearrange this as follows:

$$\frac{1}{(n - N_L)} \sum_{h=1}^{L} \frac{W_h^2 S_h^2}{a_h} = V(\bar{y}_{st}) + \sum_{h=1}^{L} \frac{W_h S_h^2}{N} \tag{7.5}$$

We then solve for $n - N_L$:

$$n - N_L = \frac{\sum_{h=1}^{L} W_h^2 S_h^2 / a_h}{V(\bar{y}_{st}) + \sum_{h=1}^{L} W_h S_h^2 / N} \tag{7.6}$$

and finally solve for n:

$$n = N_L + \frac{\sum_{h=1}^{L} W_h^2 S_h^2 / a_h}{V(\bar{y}_{st}) + \sum_{h=1}^{L} W_h S_h^2 / N} \tag{7.7}$$

This can also be rewritten using $N_L = NW_L$ and $Var(\bar{y}_{st}) = \bar{Y}^2 c^2$, where c is the target coefficient of variation (Rivest 2002):

$$n = NW_L + \frac{\sum_{h=1}^{L} W_h^2 S_h^2 / a_h}{\bar{Y}^2 c^2 + \sum_{h=1}^{L} W_h S_h^2 / N} \tag{7.8}$$

The Lavallée-Hidiroglou algorithm has often been associated with use of

Y–proportional power allocation (Gunning & Horgan 2004):

$$a_h = \frac{(W_h \bar{Y}_h)^p}{\sum_{h=1}^{L-1} (W_h \bar{Y}_h)^p} \tag{7.9}$$

where $0 < p < \infty$. We can then substitute this into equation (7.8) to obtain:

$$n = NW_L + \frac{\left(\sum_{h=1}^{L-1} (W_h S_h)^2 (W_h \bar{Y}_h)^{-p}\right) \left(\sum_{h=1}^{L-1} (W_h \bar{Y}_h)^p\right)}{\bar{Y}^2 c^2 + \sum_{h=1}^{L-1} W_h S_h^2 / N} \tag{7.10}$$

However for the purposes for this thesis we will only consider Neyman allocation:

$$a_h = \frac{W_h S_h}{\sum_{h=1}^{L-1} W_h S_h} \tag{7.11}$$

which results in the following:

$$n = NW_L + \frac{\left(\sum_{h=1}^{L-1} (W_h S_h)^2\right)^2}{\bar{Y}^2 c^2 + \sum_{h=1}^{L-1} W_h S_h^2 / N} \tag{7.12}$$

7.3 Implementation

The Lavallée-Hidiroglou algorithm starts with an arbitrary set of initial internal boundary points, and iteratively changes these until the sample size is minimised for a given coefficient of variation. The steps for the implementation of this algorithm can then outlined as follows (Gunning, Horgan & Keogh 2008):

Step 1: Sort the population into ascending order.

Step 2: Start with a set of strictly increasing internal boundaries $k_1 < k_2 < \ldots < k_{L-1}$

Step 3: Calculate stratum weight W_h, stratum mean $\bar{Y}_h$, and stratum variance S_h^2 for each of these strata.

Step 4: Replace the initial set of boundaries by solving the derivate of equation (7.8) with respect to k_h for each k_h as follows

$$\frac{\partial n}{\partial k_h} = 0 \tag{7.13}$$

Step 5: Repeat steps 3 and 4 with the new sets of boundary points, continuing until two consecutive sets are either identical or differ by negligible quantities.

Detlefsen & Veum (1991) attempted to apply the Lavallée-Hidiroglou algorithm using Neyman allocation; however found that the algorithm was often slow to converge or simply did not converge on a solution, and found different starting values resulted in sometimes substantially different resulting boundaries. We likewise found similar problems in applying this algorithm using Neyman allocation, and in particular found considerable differences in the results depending on the selection of starting values. This consequently resulted in dropping this particular version of implementing the Lavallée-Hidiroglou algorithm, and instead consideration of the implementation of the algorithm using other search methods.

The problems with the implementation of the Lavallée-Hidiroglou algorithm led Kozak (2004) to proposed an alternative implementation using

a random search method based on the work of Lednicki & Wieczorkowski (2003). The random search algorithm again minimises the sample size for a given coefficient of variation, and can be specified as follows:

Step 1: Sort the population into ascending order.

Step 2: Start with a set of strictly increasing internal boundaries $k_1 < k_2 < \ldots < k_{L-1}$

Step 3: Calculate the overall sample size n given these initial stratum boundaries.

Step 4: Generate a new point k_i' for one internal stratum boundary k_i by changing it as follows:

$$k_i' = k_i + j \tag{7.14}$$

where j is a random integer, $j \in \langle -p; -1\rangle \cup \langle 1; p\rangle$, and p is an integer selected according to the size of the population.

Step 5: Replace the boundary k_i with the new stratum boundary k_i and calculate the new overall sample size n'

Step 6: If the new sample size is less than the previous sample size then replace the existing stratum boundary k_h with the new stratum boundary k_i'

Step 7: Finish the algorithm after a given number of iterations (for example we use a maximum of $10,000$ iterations), or if the sample

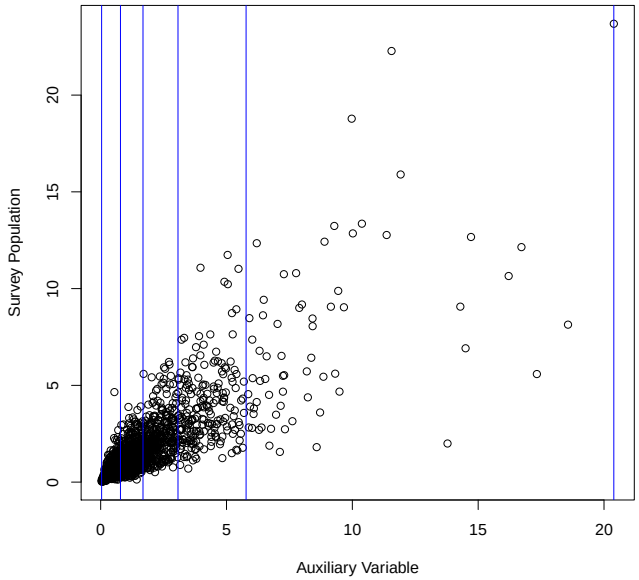

Figure 7.1: Placement of stratum boundaries using the Lavallée-Hidiroglou algorithm for a Simulated Bivariate Log-normal population ($N = 2000$)

size has not decreased in a given number of iterations (for example 100 iterations)

The initial boundary points can be a random set of increasing values between the upper and lower bounds; however better results are often obtained by first applying one of the simple algorithms mentioned in previous chapters, such as the cumulative square root of frequency rule.

The value of p denotes the upper and lower bounds on the value of j, and should be determined according to the size of the population in order to ensure the algorithm does not stop at a local minimum. Kozak (2004) suggested the value of j should not be bigger than 5 and must be greater than one, settling on a value of 3 in their analysis (which we likewise use in our results).

The above procedure can also be applied to the problem of minimising the variance of the estimates using the equation (7.7) for a given total sample size. We illustrate the results from this for the simulated bivariate log-normal population in figure 7.1, and use this approach in the next section to compare this algorithm with results from the cumulative square root of frequency and Ekman algorithms.

7.4 Results

We implement the Lavallée-Hidiroglou algorithm using the random search algorithm of Kozak (2004) on the populations in chapter 2 of this thesis, and compare this to the results from some of the other algorithms from previous chapters in table 7.1. This table shows that the Lavallée-Hidiroglou outperforms the other algorithms in the stratification of almost all of the auxiliary variable populations and auxiliary variables of the survey populations. The only exceptions are cumulative square root of frequency algorithm outperforming the Lavallée-Hidiroglou on the Sweden Municipality real estate population and the Ekman algorithm producing better (lower variance) results for the US cities population.

We also apply the Lavallée-Hidiroglou to the simulated bivariate log-normal population in figure 7.2 for different correlations between the auxiliary information and survey population. The variance of estimates for the Lavallée-Hidiroglou algorithm are similar to other results for higher correlations, however the Lavallée-Hidiroglou algorithm has consistently higher variance of estimates for lower values of the correlation between the auxil-

Table 7.1: Design effect of estimates using the Lavallée-Hidiroglou algorithm, Ekman algorithm, and the Cumulative Square Root of Frequecy algorithm

Population	L-H	CSF	Linear	Ekman	Extended
AAGIS					
- Farm Area (x)	0.0003	0.0050	0.0020	0.0015	0.0015
- Beef Cattle (y)	0.3220	0.1033	0.1198	0.1557	0.1557
SHS					
- Income (x)	0.0555	0.0569	0.0562	0.0595	0.0589
- Recreation (y)	0.7369	0.7339	0.7140	0.6798	0.6770
MU284					
- Real Estate (x)	0.0174	0.0168	0.0173	0.0179	0.0179
- Taxation (y)	0.0153	0.0224	0.0201	0.0174	0.0184
Debtors	0.0099	0.0164	0.0135	0.0147	0.0147
US Cities	0.0353	0.0294	0.0275	0.0265	0.0265
US Banks	0.0293	0.0322	0.0318	0.0308	0.0308
MRTS	0.0261	0.0276	0.0267	0.0323	0.0324
Simulated LN					
- Auxiliary (x)	0.0362	0.0367	0.0364	0.0371	0.0371
- Survey Var (y)	0.2072	0.2085	0.2086	0.2027	0.2031

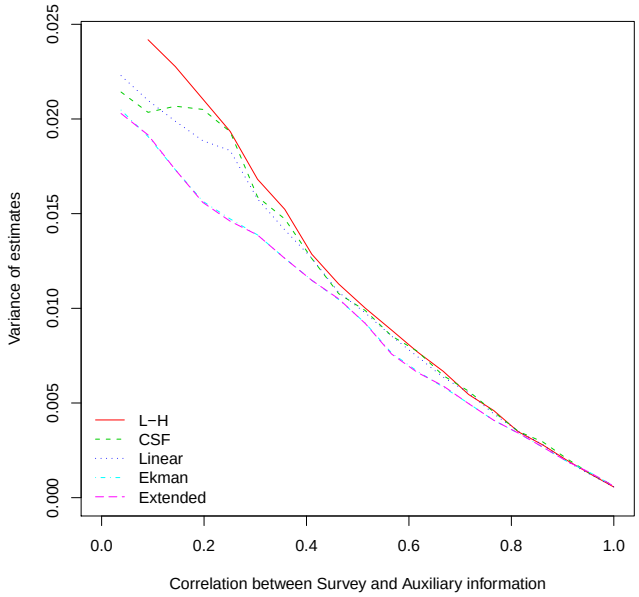

Figure 7.2: Variance of estimates using the Lavallée-Hidiroglou algorithm, Ekman algorithm, and the Cumulative Square Root of Frequecy algorithm

iary variable and survey population.

The higher variance of estimates for the Lavallée-Hidiroglou algorithm is mainly due to the implicit assumption that the stratum variance of the auxiliary variable is a good estimate of the resulting stratum variance of the survey population. This is not necessarily the case when the correlation between the auxiliary variable and the survey population is low, and consequently shows some of the limitations of a design-based approach that does not into account the strength of the relationship between the auxiliary variable and the survey population.

7.5 Summary

The Lavallée-Hidiroglou approach uses an iterative algorithm in order to estimate optimal stratum boundaries, and represents a departure from some of the previous algorithms in that is seeks to solve the equations for optimal stratum boundaries rather than derive an approximation to these boundaries. Unfortunately the original Lavallée-Hidiroglou algorithm encounters some convergence issues in obtaining an optimal solution, and can result in different solutions for differing sets of initial boundaries. This is particularly the case when applied with Neyman allocation, and has led us to abandon the original approach in favour of a random search method proposed by Kozak (2004).

This chapter has shown that the Lavallée-Hidiroglou can produces very good estimates of the optimal stratum boundaries, and our results show that the algorithm produces similar or better results than the cumulative square root of frequency and Ekman algorithms in most instances. However we have noted that the application of this algorithm to survey populations may produce poorer estimates unless we take additional steps to account for the relationship between the auxiliary and survey variables.

Chapter 8

Other Algorithms

8.1 Overview

There has been a proliferation of approximations to the solution for the optimal stratum boundaries, and the previous three chapters have gone through some of the more prominent approaches. However many of these algorithms have several shortcomings from various simplifying assumptions, or are of a complex iterative nature. This therefore makes it appropriate to consider simpler options that may provide similar results.

Cochran (1961) suggests that it may be appropriate under certain conditions to construct boundaries by simply using equal intervals on the cumulative frequency scale or equal intervals along the range of the auxiliary variable. Alternatively Gunning & Horgan (2004) suggests using a geometric progression algorithm in order to construct boundaries along the range of the variable of interest.

This chapter goes through the construction of the geometric progression,

range based, and cumulative frequency algorithms, and compares the results from such algorithms with those in the previous three chapters. We will then consider if such fast and simple algorithms could provide a reasonable approximation to the solution for the placement of optimal stratum boundaries.

8.2 Geometric Progression Algorithm

A method for estimating the optimal stratum boundary points was derived by Gunning & Horgan (2004) by assuming a uniform distribution of values within strata and equal coefficients of variation between strata. This method resulted in the construction of stratum boundaries using a simple geometric progression algorithm on the range of the auxiliary variable.

We derive the geometric progression algorithm by first setting the coefficients of variation across all strata to the same constant value c as follows:

$$c_h = \frac{S_h}{\bar{Y}_h} = c \tag{8.1}$$

We assume the distribution of values within each stratum is approximately uniform and hence obtain the following estimated value for the stratum mean:

$$\bar{Y}_h \approx \frac{k_h + k_{h-1}}{2} \tag{8.2}$$

and stratum standard deviation:

$$S_h \approx \frac{1}{\sqrt{12}}(k_h - k_{h-1}) \tag{8.3}$$

We now substitute the above values for the mean and standard deviation into equation (8.1) for the coefficient of variation as follows:

$$c_h \approx \frac{(k_h - k_{h-1})/\sqrt{12}}{(k_h + k_{h-1})/2} \tag{8.4}$$

We can then set consecutive boundaries k_h and k_{h+1} using:

$$\frac{k_{h+1} - k_h}{k_{h+1} + k_h} = \frac{k_h - k_{h-1}}{k_h + k_{h-1}} \tag{8.5}$$

as the coefficient of variation c_h is the same across all strata. We multiply by the denominators:

$$(k_{h+1} - k_h)(k_h + k_{h-1}) = (k_h - k_{h-1})(k_{h+1} + k_h) \tag{8.6}$$

and simplify to produce:

$$k_h^2 = k_{h+1}k_{h-1} \tag{8.7}$$

which is simply the product of the geometric progression identities $k_h = rk_{h-1}$ and $k_h = k_{h+1}/r$. Therefore we can specify the stratum boundaries in terms of a geometric progression:

$$k_h = ar^h \tag{8.8}$$

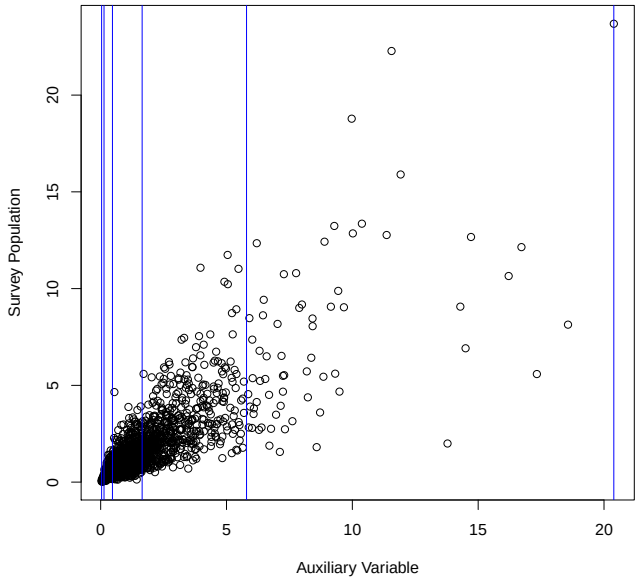

Figure 8.1: Placement of stratum boundaries using the geometric progression algorithm for a Simulated Bivariate Log-normal population ($N = 2000$)

where $a = y_1$, and:

$$r = \left(\frac{y_N}{a}\right)^{\frac{1}{L}} \tag{8.9}$$

We now implement the geometric progression algorithm using the following simple three step process:

Step 1: Sort the population into ascending order.

Step 2: Calculate the ratio r using equation (8.9) above.

Step 3: Solve equation (8.8) for each boundary.

This results in breaks set at the values $k_0 = y_1 = a, ar, ar^2, \ldots, ar^L = k_L = y_N$, as demonstrate by figure 8.1. We also implement this algorithm in the `geo` function in Appendix B.

Unfortunately there are number of issues with the geometric algorithm. Firstly the geometric algorithm requires the all values to be greater than zero. If any values are equal to or less than zero, then the value of $a = y_1$ would be equal to or less than zero and hence result in internal stratum boundaries on or outside of the upper or lower bounds of y_1 and y_N. Secondly the boundaries can change if there is a single observation added to the population that is greater than or less than the minimum or maximum values respectively. And finally the boundaries can change if there is a movement in the location of the y-axis, such as through instead specifying values as a number above or below a particular value, due to the reliance on the start value through $a = y_1$.

8.3 Range Based Approach

One of the simplest approaches to stratification is through the construction of stratum boundaries using equal intervals along the range of the auxiliary variable. Such an approach can also help ascertain the improvement in variance from the adoption of an optimal stratification design through providing a useful comparison to the optimal stratification algorithms in the previous chapters.

We can construct equal intervals along the range of the auxiliary variable as follows:

$$k_{h+1} - k_h = \frac{(y_N - y_1)}{L} \tag{8.10}$$

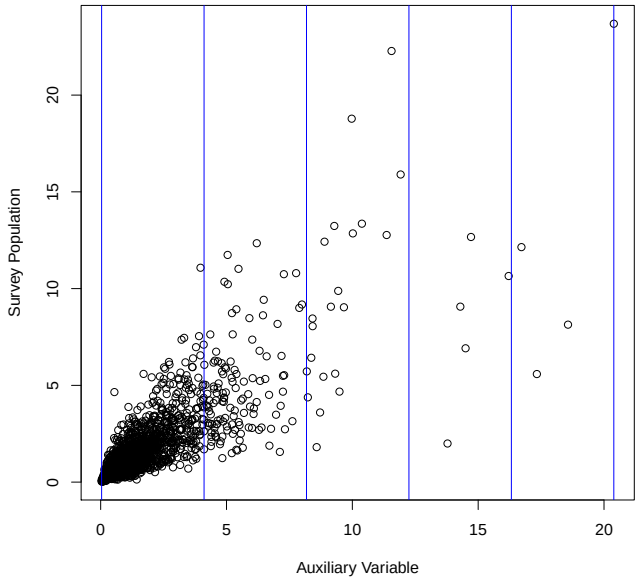

Figure 8.2: Placement of stratum boundaries using equal intervals along the range of a Simulated Bivariate Log-normal population ($N = 2000$)

This then results in the stratum boundary points of:

$$k_h = \frac{h(y_N - y_1)}{L} + y_1 \tag{8.11}$$

The implementation of such an algorithm is straightforward, and can be specified as follows:

Step 1: Calculate the range of the auxiliary variable $(y_N - y_1)$.

Step 2: Divide the result by the number of strata (as shown in (8.10) above).

Step 3: Apply equation (8.11) to obtain the stratum boundary points.

The results of this algorithm are illustrated in 8.2, and are implemented in the `eqint` function in Appendix B.

The range based approach also suffers from one of the same issues as the geometric progression algorithm, in that the addition of a value greater than y_N or less than y_1 will result in some movement of all stratum boundaries. However this is a relatively minor issue, and the addition of such values has an impact on several of the other stratification algorithms considered in this thesis.

8.4 Cumulative Frequency Approach

A common theme through several of the algorithms for the construction of stratum boundaries in this thesis is the use of intervals on some format of the cumulative frequency scale, or some combination or variant of this scale. A straightforward approach to this is to simply construct strata using equal intervals on the cumulative frequency scale, resulting in the same number of population units in each stratum.

We can calculate the stratum weight for equal intervals on the cumulative frequency scale as follows:

$$W_h = \frac{N}{L} \tag{8.12}$$

This then results in stratum boundary points:

$$F(k_h) = \frac{hN}{L} \tag{8.13}$$

We can then implement the cumulative frequency approach through the following simple algorithm:

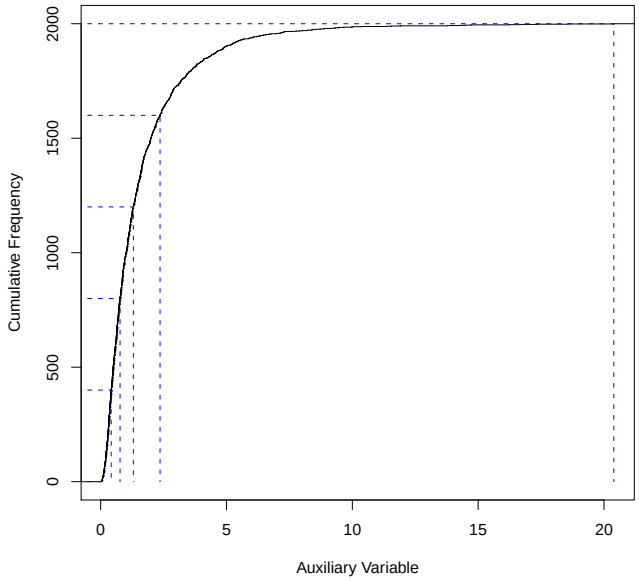

Figure 8.3: Construction of stratum boundaries using equal intervals on the cumulative frequency of a Simulated Bivariate Log-normal population ($N = 2000$)

Step 1: Sort the population in ascending order.

Step 2: Calculate the stratum boundary points on the cumulative frequency scale using equation (8.13) above.

Step 3: Find the auxiliary variable value corresponding to the value on the cumulative frequency scale.

Figure 8.3 shows the calculation of equal intervals on the cumulative frequency of the population, and then the extrapolation of these in order to find the stratum boundaries. This algorithm is also implemented in the `cfreq` function in Appendix B.

In the next section we compare the cumulative frequency algorithm with the geometric progression algorithm and the range based algorithm given in

this chapter, and then compare these with some of the other algorithms given in this thesis.

8.5 Results

We can compare the results from the cumulative frequency algorithm, the range based algorithm, and the geometric progression algorithm using the populations from chapter 2, and summarise the results of this in table 8.1. This shows that the range based and cumulative frequency algorithms provide some improvements in variance compared to simple random sampling, but provide poor estimates for optimal stratum boundaries when compared to the likes of the cumulative square root of frequency and Lavallée-Hidiroglou algorithms. Several of the adjoining stratum boundaries also needed to be collapsed for the range based algorithm as there were insufficient observations (less than two values) in the relevant strata in order to produce estimates of the stratum variance.

The geometric progression approach produces some reasonable results for highly skewed populations, and, in particular, the estimates for the AAGIS farm area, debtors, US cities, and US banks populations are comparable to estimates from the cumulative square root of frequency and Lavallée-Hidiroglou approaches. This may mean that the geometric progression algorithm is appropriate in certain circumstances where a population has a particular skew distribution, although the algorithm was unable to produce results for the Survey of Household Spending population as values were not strictly greater than zero (a requirement of the geometric approach).

Table 8.1: Design effect of estimates using the Cumulative Frequency algorithm, the Range Based algorithm, and the Geometric progression algorithm (* denotes collapsed strata due to insufficient observations, and ** denotes a population that violates the assumptions of the algorithm)

Population	Freq	Range	Geometric	CSF	L-H
AAGIS					
- Farm Area (x)	0.1061	0.1651*	0.0020	0.0050	0.0003
- Beef Cattle (y)	0.6445	0.2223*	0.8078	0.1033	0.3220
SHS					
- Income (x)	0.0862	0.5411	**	0.0569	0.0555
- Recreation (y)	0.9726	0.9126	**	0.7339	0.7369
MU284					
- Real Estate (x)	0.1408	0.2699*	0.0285	0.0168	0.0174
- Taxation (y)	0.1728	0.1158*	0.0431	0.0224	0.0153
Debtors	0.0905	0.2322	0.0175	0.0164	0.0099
US Cities	0.0704	0.0891	0.0266	0.0294	0.0353
US Banks	0.0660	0.0606	0.0293	0.0322	0.0293
MRTS	0.1123	0.3623*	0.0930	0.0276	0.0261
Simulated LN					
- Auxiliary (x)	0.0867	0.2524	0.0820	0.0367	0.0362
- Survey Var (y)	0.3480	0.5141	0.2326	0.2085	0.2072

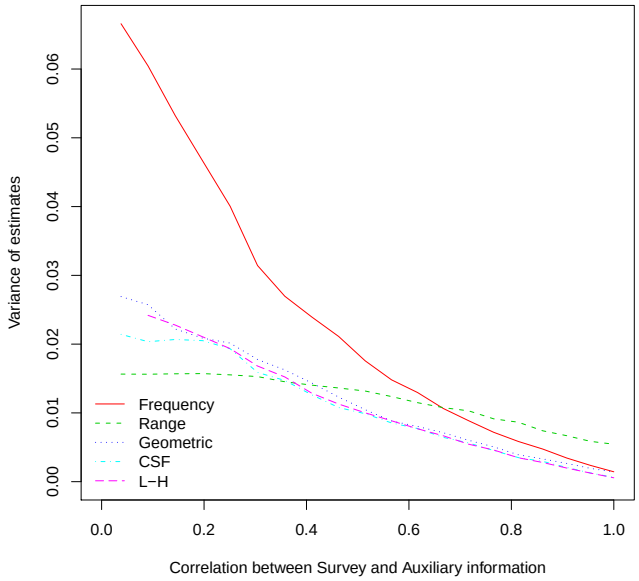

Figure 8.4: Variance of estimates using the Cumulative Frequency algorithm, the Range Based algorithm, and the Geometric progression algorithm

We further consider these algorithm in figure 8.4 using the variance of estimates for differing correlations between the auxiliary information and survey population. This shows that the range based algorithm produces reasonable estimates when there is a very low correlation between the auxiliary information and survey population, but that otherwise both the range and frequency algorithms produce poor results. The geometric progression produces results that are far closer to those of the other algorithms in this thesis, and therefore may provide a simple method to quickly estimate stratum boundaries for highly skewed populations.

8.6 Summary

This chapter has briefly considered some alternatives to the algorithms for optimal stratum boundaries covered in the previous chapters. In particular it has investigated algorithms that construct equal intervals on the cumulative frequency and range of the population, and also an algorithm that uses a geometric progression to construct stratum boundaries.

All algorithms have produced some improvements from simple random sampling, however the cumulative frequency and range based algorithms produced poor results when compared to some of the other algorithms covered in this thesis. The geometric progression algorithm produced some results that are comparable to some of the other algorithms in this thesis, and could potentially represent a very quick method of stratifying highly skewed populations.

However the geometric algorithm also has a number of limitations, and was unable to produce results when population values were not strictly greater than zero. This perhaps makes this algorithm only applicable to the stratification of a limited class of populations, and may require some further consideration to determine the appropriate situations in which to apply this algorithm.

Chapter 9

Number of Strata

9.1 Overview

The final issue to consider in the construction of an optimal stratification design is the number of strata to use. This is probably also one of the first issues that need to be considered in an optimal stratification design; however, as mentioned in chapter 3, such problems can be dependant on the approaches that will then be taken in the construction of stratum boundaries and the allocation of sample units among the strata. As such, we have left consideration of this issue until after we have investigated the allocation and boundary problems of the previous chapters.

Cochran (1977) discusses the effect of the number of strata on the variance of estimators using the cumulative square root of frequency rule, and suggests that there is little benefit from having more than six strata unless the correlation between the auxiliary information and the survey population is greater than 0.95. Kozak (2006) revisits this problem using the Lavallée-Hidiroglou

approach in the stratification of highly skewed agricultural populations and suggests that there may be some benefit from selecting a much higher number of strata.

This chapter briefly considers the analysis by Cochran (1977) and Kozak (2006), and the decrease in variance from an increase in the number of strata. We then apply this to the populations used in this thesis, and in particular through using the simulated bivariate log-normal population in chapter 2 for a variety of correlations between the auxiliary information and survey population.

9.2 Theory

We begin our consideration of the optimal number of strata by assuming that the finite population correction is negligible, and the distribution of values is approximately uniform. The uniform distribution is considerably different from the highly skewed populations observed in business and agricultural populations; however Cochran (1977) found that these assumptions still resulted in reasonable approximations for the decrease in variance from an increase in the number of strata.

We can denote the range of the distribution of values $[y_0, y_N]$ as $d = y_N - y_0$, and hence the variance of the distribution as $S^2 = d^2/12$. We can therefore calculate the variance of the sample mean for a simple random

sample of size n as:

$$\begin{aligned} V(\bar{y}) &= \frac{S^2}{n} \\ &= \frac{d^2}{12n} \end{aligned} \tag{9.1}$$

If we create L strata of equal size, then we can use the result from equation (8.10) in chapter 8 to calculate the stratum variance as $S_h^2 = d^2/12L^2$. We also notice that $W_h = 1/L$, and hence use the results from equation (3.34) from section 3.2 in order to derive:

$$\begin{aligned} V(\bar{y}_{st}) &= \frac{1}{n}\left(\sum_{h=1}^{L} W_h S_h\right)^2 \\ &= \frac{1}{n}\left(\sum_{h=1}^{L} \frac{1}{L}\frac{d}{\sqrt{12}L}\right)^2 \\ &= \frac{1}{n}\left(\frac{d}{\sqrt{12}L}\right)^2 \\ &= \frac{d^2}{12nL^2} \\ &= \frac{V(\bar{y})}{L^2} \end{aligned} \tag{9.2}$$

This suggests that the variance of the sample mean is inversely proportional to the square of the number of strata. However this result includes several strong assumptions, and does not consider the relationship between the auxiliary information and the survey population.

We can briefly extend the above results, through a simple extension to the design-based approach of the thesis, by considering a linear relationship

between the survey population and auxiliary information as follows:

$$y = \alpha + \beta x + e \tag{9.3}$$

We then use equation (2.22) of chapter 2, ignoring the finite population correction, and substituting in $n_h = n/L$:

$$\begin{aligned} V(\bar{y}_{st}) &= \sum_{h=1}^{L} \frac{W_h^2 S_{yh}^2}{n_h} \\ &= \frac{L}{n} \sum_{h=1}^{L} \frac{1}{L^2} \frac{S_y^2}{L^2} \\ &= \frac{S_y^2}{nL^2} \end{aligned} \tag{9.4}$$

We can now calculate the variance of equation (9.3) as follows (Cochran 1977):

$$\begin{aligned} V(\bar{y}_{st}) &= \frac{\beta^2 L}{n} \sum_{h=1}^{L} W_h^2 S_{xh}^2 + \frac{L}{n} \sum_{h=1}^{L} W_h^2 S_e^2 \\ &= \frac{\beta^2 L}{n} \sum_{h=1}^{L} W_h^2 S_{xh}^2 + \frac{S_e^2 L}{n} \sum_{h=1}^{L} W_h^2 \end{aligned} \tag{9.5}$$

where S_e^2 is constant. We simply equation (9.5) using $\sum W_h^2 \geq 1/L$ as

follows:

$$\begin{aligned} V(\bar{y}_{st}) &= \frac{\beta^2 L}{n}\sum_{h=1}^{L} W_h^2 S_{xh}^2 + \frac{S_e^2 L}{n}\sum_{h=1}^{L} W_h^2 \\ &\geq \frac{1}{n}\left(\frac{\beta^2 S_x^2}{L^2} + S_e^2\right) \\ &\geq \frac{S_y^2}{n}\left[\frac{\rho^2}{L^2} + (1-\rho^2)\right] \end{aligned} \tag{9.6}$$

where ρ is the correlation between survey population and the auxiliary information. This therefore suggests that the variance is not only inversely related to the square of the number of strata, but that it is also influenced by the correlation between the auxiliary information and survey population. We consider this further in the next section through results for various correlations using a simulated bivariate log-normal population.

9.3 Applications

We examine the effect of increasing the number of strata from $L = 2$ to $L = 12$ for the various populations used in this thesis in tables 9.1 and 9.2. These tables show a decrease in variance of the estimates, relative to simple random sampling (the design effect), for the auxiliary variables of the survey populations and variables for the auxiliary variable populations from an increase in the number of strata. However these decreases are not constant, and there is a diminishing marginal return from an increase in the number of strata.

The three survey populations and one simulated population exhibit de-

Table 9.1: Design effect of estimates from changes in the number of strata using the Cumulative Square Root of Frequency rule (part 1)

	Number of strata				
Population	2	3	4	5	6
AAGIS					
- Farm Area (x)	0.0212	0.0108	0.0071	0.0050	0.0036
- Beef Cattle (y)	0.1745	0.1135	0.1088	0.1033	0.0993
SHS					
- Income (x)	0.3386	0.1570	0.0884	0.0569	0.0387
- Recreation (y)	0.7211	0.7475	0.6996	0.7339	0.7615
MU284					
- Real Estate (x)	0.2406	0.0969	0.0324	0.0168	0.0108
- Taxation (y)	0.2367	0.0889	0.0268	0.0224	0.0189
Debtors	0.1220	0.0479	0.0267	0.0164	0.0116
US Cities	0.1844	0.0902	0.0525	0.0294	0.0202
US Banks	0.1918	0.0916	0.0524	0.0322	0.0178
MRTS	0.2366	0.0965	0.0452	0.0276	0.0191
Simulated LN					
- Auxiliary (x)	0.2501	0.1114	0.0609	0.0367	0.0245
- Survey Var (y)	0.3675	0.2800	0.2357	0.2085	0.2059

creases in the variance of the estimates for between two and seven strata. However the design effect for the AAGIS Beef Cattle population actually increases when moving to eight strata, and all four populations reach a plateau at eight or nine strata (with some residual variation).

Figure 9.1 further explores the effect of increasing the number of strata on the variance of the survey estimates for a variety of correlation coefficients. This clearly shows the diminishing marginal returns from increasing the number of strata, with little benefit from increasing the number of strata when there is less correlation between the auxiliary information and the survey population.

Table 9.2: Design effect of estimates from changes in the number of strata using the Cumulative Square Root of Frequency rule (part 2)

	Number of strata					
Population	7	8	9	10	11	12
AAGIS						
- Farm Area (x)	0.0027	0.0023	0.0018	0.0015	0.0013	0.0011
- Beef Cattle (y)	0.0951	0.1000	0.0926	0.0970	0.0914	0.0943
SHS						
- Income (x)	0.0275	0.0218	0.0167	0.0137	0.0115	0.0089
- Recreation (y)	0.7467	0.6996	0.7125	0.7020	0.7335	0.6784
MU284						
- Real Estate (x)	0.0081	0.0060	0.0039	0.0032	0.0030	0.0022
- Taxation (y)	0.0142	0.0140	0.0127	0.0122	0.0129	0.0131
Debtors	0.0085	0.0062	0.0050	0.0041	0.0033	0.0028
US Cities	0.0140	0.0115	0.0092	0.0069	0.0058	0.0049
US Banks	0.0170	0.0118	0.0096	0.0079	0.0067	0.0057
MRTS	0.0144	0.0109	0.0088	0.0070	0.0059	0.0048
Simulated LN						
- Auxiliary (x)	0.0171	0.0134	0.0107	0.0086	0.0067	0.0061
- Survey Var (y)	0.2029	0.1832	0.1852	0.1811	0.1820	0.1831

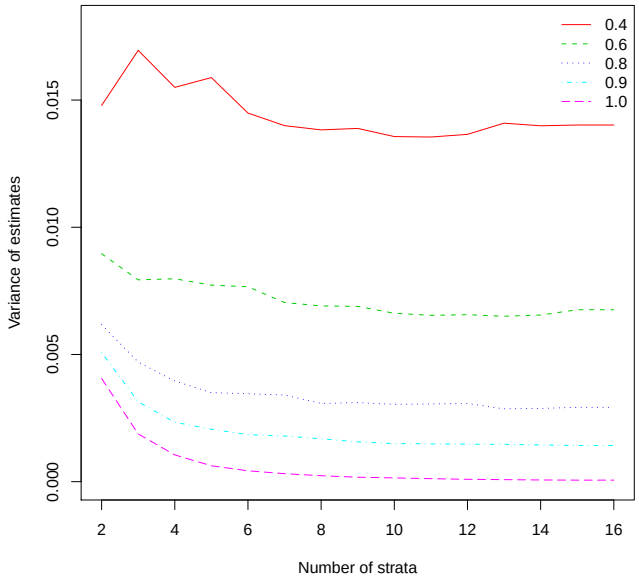

Figure 9.1: Variance of estimates from changes in the number of strata using the Cumulative Square Root of Frequency rule

Overall we see that there may be some benefit from more than six strata when the correlation is above $\rho = 0.6$, and there may be some instances where ten or more strata may be useful if the correlation is greater than 0.9. However we have ignored the cost of increasing the number of strata from such considerations, and any increase in cost would need to be taken into account when considering the decrease in variance from a higher number of strata. This alone may result in fewer strata being selected, and we have attempted to take this into consideration throughout this thesis through using only five strata.

9.4 Summary

This chapter has briefly considered the optimal number of strata to construct in order to support an optimal stratification design. We have examined the decrease in variance from an increase in the number of strata, and estimated this as approximately proportional to the inverse of the square of the number of strata. We have then seen that this results in diminishing marginal returns from increases in the number of strata.

We have also undertaken a brief foray into model assisted approaches through constructing a simple linear relationship between the survey population and the auxiliary information. This has allowed us to demonstrate that a decrease in variance from an increase in strata is also related to the correlation between the survey and auxiliary information, and we have likewise observed this through decreases in the variance of estimates for simulated populations with various correlations between the survey and auxiliary information.

Overall this chapter has observed that there are considerable gains from increasing the number of strata up to five or six strata, and may still be some gains from more than six strata when the correlation between the auxiliary information and survey population is very high. However the cost of constructing larger numbers of strata needs to be taken into consideration, and may therefore support the original suggestion of around five or six strata for even highly correlated populations.

Chapter 10

Discussion

10.1 Overview

This thesis has investigated univariate design-based optimal stratification as applied to highly skewed populations, such as those present in business and agricultural surveys. In particular we have considered algorithms and rules to address the three problems identified in section 1.3 of chapter 1:

- The number of strata that should be used
- The construction and placement of stratum boundaries
- The allocation of sample units among the strata

This discussion briefly considers and summarises the work undertaken in this thesis in addressing these questions, and in particular the work on the construction and placement of optimal stratum boundaries. It also summarises the results from the application of the various approaches from the various actual and simulated populations used in this thesis.

The next section of this discussion goes through the work in the first three chapters, and considers the theory underpinning stratification and the results from optimal allocation. The third section examines the work on optimal stratum boundaries, and briefly considers the work on the optimal number of strata. Finally we then consider the limitations of this work and areas for further development in the future.

10.2 Optimal Stratification

The first and second chapters of this thesis provides a considerable background to stratification, and considers a number of business and agricultural populations that are used throughout this thesis. Within this we provide equations for the estimation of the variance of simple random sampling given a stratified sampling design, and use these extensively for the calculation of the design effect of the various optimal stratification algorithms and rules throughout this thesis.

Chapter 3 considers the issue of the number of units to sample from each stratum, and derives the equations for the optimal allocation of a sample among the strata. We find that proportional allocation always results in values for the variance of the estimator that are less than or equal to that of simple random sampling, and likewise optimal allocation generally results in lower variance estimates than proportional allocation and simple random sampling.

However we find that optimal allocation can produce higher variance estimates when the correlation between the auxiliary information and the

survey population is low (in our case less than 0.4). We ultimately deduce that this is due to the stratum variance of the auxiliary information being used as an estimate of the stratum variance of the survey population, despite a low correlation between these two variables, and suggest the correlation between the two variables needs to be taken into consideration in an optimal stratification design through possibly constructing a model of the relationship between the auxiliary variable and survey population.

10.3 Optimal Boundaries

The construction of optimal stratum boundaries has been a core component of the work in this thesis, and we derive the equations for optimal boundaries in chapter 4. However these equations are difficult to solve for more than two strata, and we therefore present a number of approximate methods in chapters 5 to 8

The cumulative square root of frequency was one of the first approximations to the intractable equations for optimal stratum boundaries, and we derive and implement this algorithm in chapter 5. This algorithm relies on the construction of initial intervals to calculate the cumulative square root of frequency scale, and we unfortunately find the number of initial intervals can have a considerable impact on the variance of the estimators. This leads us to conclude that the number of initial intervals needs to be sufficiently large to ensure that there are a reasonable number of points to approximate the equal intervals on the cumulative square root of frequency scale, but not so large as to result in changes in the placement of boundaries (finding that

a number of initial intervals equal to around ten times the number of strata as suitable for the purposes of this thesis).

We extend the cumulative square root of frequency rule to address some of the variation in stratum boundaries due to the construction of initial intervals, through proposing linear and spline interpolation extensions to the cumulative square root of frequency algorithm. We find that both of these extensions result in improvements to the cumulative square root of frequency rule, and suggest using the linear interpolation variant due to the ease of implementation.

The Ekman algorithm is an alternative approach to the cumulative square root of frequency rule, and we derive and implement this algorithm in chapter 6. We also present an extended approach based on the work of Hedlin (2000) and a new kernel density based algorithm, and then consider several important results relating to the construction of boundaries using the Ekman algorithm. Finally we compare the three Ekman based algorithms and find very similar results, with the algorithms generally resulting in variance of estimates that are equalivalent or slight improvements on the estimates from the cumulative square root of frequency rule. We also note the benefits of the extended approach and kernel density approach in estimation optimal boundaries for very small populations, and suggest the adoption of the extended approach in such circumstances due to the speed of this algorithm compared to the kernel density based variant.

We also look at the Lavallée-Hidiroglou algorithm, and find that it is often slow or does not converge for Neyman allocation. This leads us to adopt a random search model from Kozak (2004), and we provide the background

theory and implementation details of this algorithm in chapter 7. This algorithm is then applied to the same populations as preivous algorithms, and we find that the Lavallée-Hidiroglou algorithm generally produces superior results for the stratification of auxiliary variable populations and auxiliary variables of survey populations in this thesis. This however changes for the survey populations, and we note that this is more than likely due to such algorithms not taking into consideration the relationship between the auxiliary variable and the survey population.

We finally consider several alternatives to these boundary algorithms, and found that a geometric progression algorithm may provide quick estimates for stratum boundaries for some highly skewed populations. However this approach also has a number of limitations that may make less useful than some of the other methods.

We have also shown that there are considerable gains from increasing the number of strata up to six strata, and often further gains for more than six strata when there is a sufficiently high correlation between the auxiliary variable and survey population. However we have not considered the cost of constructing such strata, and suggest the gains from more than five or six strata may be offset by the increased cost of the survey design.

10.4 Summary

This thesis has provided an in depth investigation to optimal stratification of highly skewed populations, similar to those encountered in business and agricultural populations, and has in particular focused on algorithms to con-

struct and estimate the placement of optimal stratum boundaries. Within this we have considered the assumptions that underpin various existing algorithms, and presented a number of new and modified extensions to these algorithms.

Overall we find that most boundary algorithms provide good results, and, of these, the Lavallée-Hidiroglou algorithm consistently produces better (lower variance) estimates. The Lavallée-Hidiroglou can however be unstable, and is slightly more complicated that some of the other algorithms. We therefore also suggest that the extended Ekman algorithm may be appropriate in order to produce more stable results, or possibly the proposed linear interpolation extension to the cumulative square root of frequency rule in order to avoid the computational complexity of an iterative algorithm.

We have deliberately taken a univariate design-based approach to optimal stratification in this thesis in order to limit the size of the work, and have also assumed the auxiliary variable matches or is a close approximation to the survey population of interest. This provides considerable scope for further development of optimal stratification algorithms through moving beyond a design-based approach, taking account of the relationship between the auxiliary variable and survey population, and also through extending algorithms into a multivariate environment.

There is also scope for further development of the algorithms presented in this thesis, particularly through amendments and extensions to the existing approaches. Many of the algorithms make strong assumptions relating to the distribution of population values, and there is room to improve the iterative process of several algorithms in order to produce faster and more consistent

results.

There are also numerous other algorithms for the allocation of units among the strata. Other alternatives such as power allocation have been used by other authors, and the investigation of such algorithms could result in further developments in the algorithms for optimal stratification.

Appendix A

Populations

A.1 Overview

This population appendix provides information on the datasets used in this thesis. Further information on these populations, and in particular information on the construction of the simulated bivariate log-normal population, is given in section 2.6 of chapter 2.

A.2 Australian Agricultural and Grazing Industries Survey

The Australian Agricultural and Grazing Industries Survey (AAGIS) dataset is a collection of $1,652$ Australian broadacre farms sampled in the Australian Agricultural and Grazing Industries Survey used in Chambers (1996) and Karlberg (2000). There are sixteen variables in this dataset, relating to financial and production aspects of the operations of each farm over the year

of the survey, and our primary interest will be in the farm area and beef cattle series.

The sixteen variables in the Australian Agricultural and Grazing Industries Survey dataset are as follows:

- Id: Unique identifier for each sample farm
- State: The State in which the farm is situated (1 = New South Wales, 2 = Victoria, 3 = Queensland, 4 = South Australia, 5 = Western Australia, 6 = Tasmania, 7 = Northern Territory).
- Zone: The climatic zone in which the farm is situated (1 = low rainfall, 2 = medium rainfall, 3 = high rainfall).
- Region: Within-state regions (29), corresponding to different farming areas (nested within State and Zone) in which the farm is situated.
- Industry: The industrial classification of the farm (1 = crops specialist, 2 = mixed livestock and crops, 3 = sheep specialist, 4 = beef specialist, 5 = mixed livestock).
- Weight: Sample weight for the farm.
- Equity (A$): Farm operator's equity (value of farm business - farm debt).
- FarmDebt (A$): Total debt of the farm business.
- TCC (A$): Total Cash Costs of the farm business over the surveyed year.

- TCR (A$): Total Cash Receipts of the farm business over the surveyed year.
- FCI (A$): Farm Cash Income = TCR-TCC
- FarmArea (thousands of hectares): Total area of the farm.
- CropsArea (hectares): Area of crops grown on the farm.
- BeefCattle (thousands): Number of beef cattle on the farm
- Sheep (number): Number of sheep on the farm
- DSE: Overall measure of farm size (Dry Sheep Equivalent), defined as number of sheep + 8*number of beef cattle + 12*crops area in hectares

A.3 Survey of Household Spending

The Survey of Household Spending (SHS) dataset is a collection of $16,057$ observations from the 2001 Survey of Household Spending carried out by Statistics Canada, and is sourced from the `stratification` package of the `R` programming language. The dataset contains seven variables relating to household spending during the reference year, and we will mainly use the household income and household spending variables.

The seven variables in the Survey of Household Spending dataset are as follows:

- CASEID: Identification number
- WEIGHT: Weight at household level

- PROVINCP: Province or territory code
- URBRUR: Urban rural code
- URBSIZEP: Size of area of residence code
- HHINCTOT: Household income before taxes (thousands of dollars)
- M101: Household spending on recreation (thousands of dollars)

A.4 Sweden Municipalities

The Sweden Municipalities (MU284) dataset is a collection of information on 284 Swedish Municipalities from Särndal, Swensson & Wretman (1992) and reproduced in the `sampling` package of the `R` programming language. The dataset contains eleven variables, and we are primarily interested in the real estate and taxation revenue variables

The complete list of the eleven variables in the Sweden Municipalities dataset are as follows:

- LABEL: Identifier running from 1 to 284
- P85: 1985 population (in thousands)
- P75: 1975 population (in thousands)
- RMT85: Revenues from the 1985 municipal taxation (in millions of kronor)
- CS82: Number of Conservative seats in municipal council

- SS82: Number of Social-Democratic seats in municipal council
- S82: Total number of seats in municipal council
- ME84: Number of municipal employees in 1984
- REV84: Real estate values according to 1984 assessment (in millions of kronor)
- REG: Geographic region indicator
- CL: Cluster indicator (a cluster consists of a set of neighbouring municipalities)

A.5 Auxiliary Variable Populations

The auxiliary variable populations in this thesis are a collection of four univariate populations (of an auxiliary variable assumed to be the same as the survey population) sourced from the `stratification` package in `R`. The `stratification` package is predominately an implementation of the Lavallée-Hidiroglou algorithm given in chapter 7, and many of these populations have been used in the past to implement some of the stratum boundary algorithms in this thesis.

A.5.1 Debtors

The Debtors dataset is an accounting population of debtors in an Irish firm used in Horgan (2003). There are $3,369$ observations in the dataset, and the values in the dataset are between 40 and $28,000$.

A.5.2 US Cities

The US Cities dataset is list of the population in thousands of US cities in 1940 used in Cochran (1961) and Gunning & Horgan (2004). There are $1,038$ observations in the dataset, and the dataset contains values between 10 and 198 thousand people.

A.5.3 US Banks

The US Banks dataset is a list of the resources in millions of dollars of large commercial US banks used in Cochran (1961) and Gunning & Horgan (2004). There are 357 observations in the datasets, and values range between 70 and 997 million dollars.

A.5.4 Monthly Retail Trade Survey

The Monthly Retail Trade Survey (MRTS) dataset is a simulation using a skew-t distribution of the size measure for Canadian retailers in the Monthly Retail Trade Survey of Statistics Canada and used in Baillargeon, Rivest & Ferland (2007). The size measure is created using a combination of independent survey data and three administrative variables from the corporation tax return, and the resulting dataset contains $2,000$ observations between the values of $1,412$ and $486,400$ (the values have been divided by $1,000$ for the purposes of this thesis).

Appendix B

Source Code

B.1 Overview

A significant number of functions have been construction in the `R` programming language to both implement and extend the various boundary algorithms in (Dalenius & Hodges 1957), (Ekman 1959*a*), (Lavallée & Hidiroglou 1988), (Cochran 1961), (Horgan 2006), and others. The code relating to the implementation of these algorithms is presented in this appendix to provide further details on the implementation of these algorithms, and provide a reference for any others in the future that may wish to implement such algorithms in the `R` programming language.

The next section covers functions to implement the various boundary algorithms covered in chapters 5, 6, and 8 (with the Lavallée-Hidiroglou algorithm implemented using the `stratification` package in `R`, as discussed in chapter 7). The third section covers the allocation algorithms in chapter 3, and functions to sample and summarise the results of the various algorithms.

The final two sections include several complementary programs constructed to enhance the base functionality of the R programming language for the purposes of the above algorithms, and a function to generate log-normal random variables with a specified correlation coefficient (as discussed in chapter 2).

B.2 Boundary Algorithms

B.2.1 Cumulative Square Root

```
#########################################################################
# R Function: csf
#
# Purpose:
#   Estimates optimal boundaries for stratified sampling using the
#   cumulative square root of frequency rule.
#
# Define variables:
#   Input:
#     x         - Raw data
#     strata    - Number of strata
#     intervals - Number or vector of interval boundaries for the
#                 calculation of cumulative frequencies
#     freq      - Vector of frequencies for the given interval
#                 boundaries
#     method    - Boundary method
#     plot      - Option to plot results
#     ...       - Further arguments for plot
#   Output:
#     r         - Stratification information (of class histogram)
#
# Notes:
#   Allows step, linear, and spline interpolation of boundaries.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/02/2009  M. Hayward      Original code
#   14/03/2009  M. Hayward      Update name & remove data field
#   13/09/2009  M. Hayward      Added boundary functions
#
```

```
csf <- function(x, strata = 5, intervals = strata * 10, freq = NULL,
    method = c("stepfun", "approxfun", "splinefun"),
    plot = FALSE, ...){

    ######################################################################
    # Set up intervals and calculate frequencies

    # Check number of strata
    if (length(strata) > 1 || !is.numeric(strata) ||
        !is.finite(strata) || strata < 1)
        stop("invalid number of 'strata'")

    # Obtain the unevaluated expression for x and turn into a string
    xname <- paste(deparse(substitute(x), 500), collapse = "\n")

    # Calculate freq if missing
    if (missing(freq)){

        # Set up vector of intervals
        if (length(intervals) > 1) {
            # Check the number of strata and intervals
            if (length(intervals) < strata)
                stop("invalid number of 'strata' or 'intervals'")
            # If vector of intervals then sort
            intervals = sort(intervals)
        } else {
            # Check intervals value (must be greater than strata)
            if (!is.numeric(intervals) || !is.finite(intervals) ||
                intervals < strata)
                stop("invalid number of 'intervals'")
            # Create vector of intervals
            intervals <- seq(min(x),max(x),length.out=intervals+1)
        }

        # Calculate frequencies (checks population is numeric etc)
        freq <- bins(x,breaks=intervals)$counts

    # Check vector of freq if specified
    } else {

        # Check the number of strata and intervals
        if (length(strata) > length(intervals))
            stop("invalid number of 'strata' or 'intervals'")

        # Check the number of intervals and freq
        if (length(freq) != length(intervals) - 1)
            stop("invalid number of 'intervals' or 'freq'")
```

```
    # Check intervals are sorted
    if (is.unsorted(intervals))
        stop("'intervals' are not sorted")

    # Set xname if missing (could be missing if freq specified)
    if (nchar(xname) < 1)
        xname <- "x"

}

#########################################################################

#########################################################################
# Find actual and theoretical boundaries

# Calculate cumulative square root of frequencies (cs)
cs <- c(0,cumsum(sqrt(freq)))

# Calculate theoretical boundaries (tb) using equal intervals on
# the cumulative square root (cs) scale
tb <- seq(0,max(cs),length.out=strata+1)

#########################################################################

#########################################################################
# Find boundaries values closest to theoretical boundaries

# Match method argument
method <- match.arg(tolower(method), c("stepfun", "approxfun",
    "splinefun"))

# Find break points
xb <- switch(method,

    # Stepwise function
    "stepfun" = csf.stepfun(tb=tb,cs=cs,intervals=intervals),

    # Linear interpolation
    "approxfun" = csf.approxfun(tb=tb,cs=cs,intervals=intervals),

    # Monotonic spline interpolation
    "splinefun" = csf.approxfun(tb=tb,cs=cs,intervals=intervals,
        fn=splinefun,method="monoH.FC"),

    # Unknown method
    stop("unknown 'breaks' algorithm")

)
```

```
    ######################################################################

    ######################################################################
    # Set-up output

    if (missing(x)){

        # Calculate counts
        xc <- diff(approxfun(intervals,c(0,cumsum(freq)))(xb))

        # Construct histogram object
        r <- as.hist(breaks=xb,counts=xc,xname=xname)

    } else {

        # Construct histogram object
        r <- bins(x,breaks=xb)

        # Correct xname
        r$xname <- xname

    }

    ######################################################################

    ######################################################################
    # Output results

    # Output graphs for csf function
    if (plot){

        # Histogram of initial breaks and final boundaries
        # Plot function of csf versus boundaries
        csf.plot(final=r,initial=as.hist(intervals,freq,xname),
            method=method,...)

        # Return results
        invisible(r)

    } else {

        # Return results
        r

    }

    ######################################################################

}
```

```
#########################################################################
```

```
#########################################################################
# R Function: csf.stepfun
#
# Purpose:
#   Find closest points to the theoretical breaks on cumulative square
#   root of frequency scale.
#
# Define variables:
#   Input:
#     tb        - Theoretical breaks (on CSF scale)
#     cs        - Cumulative square root of frequency scale
#     maxit     - Maximum number of iterations
#   Output:
#     xb        - Stratificaiton boundaries
#
# Notes:
#   Adjusts for duplicate values, and minimises the variance of the
#   distance on the CSF scale between breaks.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   13/09/2009  M. Hayward      Original code
#

csf.stepfun <- function(tb, cs, intervals, maxit = 100){

    #####################################################################
    # Find intervals

    # Find indicies closest to theoretical boundaries (which boundary)
    wb <- findBoundary(tb,cs)

    #####################################################################

    #####################################################################
    # Resolve duplicate breaks

    # Initialise iterations
    iter = 1

    # Find any duplicate values
    while(any(d <- duplicated(wb))){
```

```
    # Test maximum iterations
    if (iter >= maxit){
        warning("'maxit' reached for duplicates")
        break
    }

    # Find the index of closest spare (potential) values
    s <- (1:length(cs))[-wb]

    # Find new boundary values from set of spare values
    # Must adjust for spare values (as cs subset by s)
    wb[d] <- s[findBoundary(tb[d],cs[s])]

    # Increment iterations
    iter = iter + 1

}

# Sort the resulting boundary indicies
wb <- sort(wb)

#########################################################################

#########################################################################
# Find improvements in breaks

# Objective function
csf.var <- function(x){
    var(diff(sort(x)))
}

# Find length of wb
lwb = length(wb)

# Initialise iterations
iter = 1

# Test possible break improvements
while (TRUE){

    # Test maximum iterations
    if (iter >= maxit){
        warning("'maxit' reached for break improvements")
        break
    }

    # Find available points
    wa <- (1:length(cs))[-wb]
```

```
    # Potential replacement points (returns index of wa)
    wa0 <- findInterval(wb,wa) # Previous values
    wa1 <- wa0 + 1 # Next values

    # Ensure valid indicies (> 0 & <= length(wa))
    wb0 <- which(wa0 > 0) # wa0 <= length(wa) implicit
    wb1 <- which(wa1 <= length(wa)) # wa1 > 0 implicit

    # Remove first and last values (1 & length(wb))
    wb0 <- wb0[wb0 > 1 & wb0 < lwb]
    wb1 <- wb1[wb1 > 1 & wb1 < lwb]

    # Combine series
    wai <- c(wa0[wb0],wa1[wb1])
    wbi <- c(wb0,wb1)

    # Find length of wai/wbi
    lwbi <- length(wbi)

    # Construct test series
    wbt <- rep.int(wb,lwbi)
    wbt[(0:(lwbi-1))*lwb+wbi] <- wa[wai]

    # Evaluate variance
    cst <- apply(matrix(cs[wbt],ncol=lwbi),2,csf.var)

    # Find minimum, and retain if lower (break otherwise)
    if (any(cst < csf.var(cs[wb]))){
        wb[wbi[which.min(cst)]] <- wa[wai[which.min(cst)]]
    } else {
        break
    }

    # Increment iterations
    iter = iter + 1

}

#########################################################################

#########################################################################
# Output result

# Sort the resulting boundaries
xb <- sort(intervals[wb])

# Return results
xb
```

```
    ######################################################################

}

##########################################################################



##########################################################################
# R Function: csf.approxfun
#
# Purpose:
#   Interpolate closest points to theoretical breaks on cumulative
#   square root of frequency scale.
#
# Define variables:
#   Input:
#     tb        - Theoretical breaks (on CSF scale)
#     cs        - Cumulative square root of frequency scale
#     intervals - Initial breaks
#     fn        - Interpolation function (default = linear)
#     tol       - Tolerance level
#     maxit     - Maximum number of iterations
#     ...       - Further arguments for fn
#   Output:
#     xb        - Stratificaiton boundaries
#
# Notes:
#   Suitable for monotonic functions based on approxfun. This includes
#   both constant and linear version of approxfun, and the monotonic
#   spline function (R 2.8.0 onwards).
#
#   Adjusts for duplicate values, and minimises the distance to
#   theoretical breaks on the CSF scale (to within value of tol).
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   13/09/2009  M. Hayward      Original code
#

csf.approxfun <- function(tb, cs, intervals, fn = approxfun,
    tol = 1e-07, maxit = 50, ...){

    ######################################################################
    # Set-up function
```

```
# Set tolerance
tol <- tol * mean(diff(cs))

# Adjust for duplicates
if (any(d <- duplicated(cs))){

    # Adjust duplicate values
    cs[d] <- cs[d] + tol

    # Remove if more than one duplicate for each value
    if (any(e <- duplicated(cs,fromLast = TRUE))){
        cs <- cs[!e]
        intervals <- intervals[!e]
    }

}

# Create function
fx <- fn(intervals,cs,...)

# Get internal theoretical breaks
tb <- tb[c(-1,-length(tb))]

#########################################################################

#########################################################################
# Find intial points

# Find indicies of initial points
wb <- findInterval(tb,cs)
x0 <- intervals[wb]
x1 <- intervals[wb+1]

# Estimate closest points
xb <- fn(cs,intervals,...)(tb)

# Check monitonicity of fn
if ((x0 > xb) || (xb > x1))
    stop("'fn' must be (weakly) monotonic")

#########################################################################

#########################################################################
# Find improvements in breaks

# Initialise iterations
iter = 1

# Look for improvements in internal breaks
```

```
    while(max(abs(fxb <- fx(xb) - tb)) > tol){

        # Test maximum iterations
        if (iter >= maxit){
            warning("'maxit' reached")
            break
        }

        # Check breaks
        wb <- (fxb < 0)

        # Update x-values
        x0[wb] <- xb[wb]
        x1[!wb] <- xb[!wb]

        # Find new mid-point
        xb <- (x0 + x1) / 2

        # Increment iterations
        iter = iter + 1

    }

    ###########################################################################

    ###########################################################################
    # Output result

    # Add first and last points
    xb <- c(min(intervals),xb,max(intervals))

    # Output results
    xb

    ###########################################################################

}

###############################################################################




###############################################################################
# R Function: csf.plot
#
# Purpose:
#   Plots diagnostic graphs for boundaries created by the cumulative
#   square root of frequency rule.
```

```
#
# Define variables:
#   Input:
#     final     - Final breaks from CSF
#     initial   - Initial bins used in CSF
#     method    - Boundary method
#     do.points - Option to plot interval points
#     main      - Main title for plots
#     xlab      - Label for x-axis
#     ylab      - Label for y-axis (CSF plot only)
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/02/2009  M. Hayward      Original code
#   13/09/2009  M. Hayward      Added boundary functions
#

csf.plot <- function(final, initial, method = "stepfun",
    do.points = TRUE, main = paste("Stratification of",final$xname),
    xlab = "Auxiliary Variable", ylab = expression(paste("Cumulative",
    phantom(0),sqrt(italic("f"))))){

    ######################################################################
    # Set-up breaks function

    # Calculate cumulative square root of frequencies (cs)
    cs <- c(0,cumsum(sqrt(initial$counts)))

    # Match method argument
    method <- match.arg(tolower(method), c("stepfun", "approxfun",
        "splinefun"))

    # Match breaks function
    fn <- switch(method,

        # Stepwise function
        "stepfun" = stepfun(initial$breaks[-1],cs),

        # Linear interpolation
        "approxfun" = approxfun(initial$breaks,cs),

        # Monotonic spline interpolation
        "splinefun" = splinefun(initial$breaks,cs,method="monoH.FC"),

        # Unknown method
        stop("unknown 'breaks' algorithm")

    )
```

```
    # Calculate values of boundaries on CSF scale
    tb <- fn(final$breaks)

    ########################################################################

    ########################################################################
    # Create histogram of initial breaks and final boundaries

    # Histogram of results (overlays initial bins with final bins)
    plot(initial,freq=FALSE,ylim=c(0,max(initial$density,
        final$density)),main=main,xlab=xlab,col=5,lty=2)
    lines(final,density=5)

    ########################################################################

    ########################################################################
    # Plot function of csf versus boundaries

    # Create new window
    x11()

    # Plot of cumulative frequencies
    if (method == "stepfun") {
        plot(fn,xlim=range(initial$breaks),ylim=range(cs),
            do.points=FALSE,main=main,xlab=xlab,ylab=ylab)
    } else {
        plot(fn,xlim=range(initial$breaks),ylim=range(cs),
            main=main,xlab=xlab,ylab=ylab)
    }

    # Add points
    if (do.points){
        points(initial$breaks,cs)
    }

    # Plot boundaries
    for(i in 1:length(final$breaks)){
        lines(c(final$breaks[1]-diff(range(initial$breaks)),
            final$breaks[i],final$breaks[i]),c(tb[i],tb[i],0),
            col=4,lty=2)
    }

    ########################################################################

}

########################################################################
```

B.2.2 Ekman Algorithm

```
#######################################################################
# R Function: ekman
#
# Purpose:
#   Estimates optimal boundaries for stratified sampling using the
#   Ekman algorithm.
#
# Define variables:
#   Input:
#     x         - Raw data
#     strata    - Number of strata
#     freq      - Vector of frequencies for the given data
#     method    - Boundary method
#     plot      - Option to plot results
#     ...       - Further arguments for plot
#   Output:
#     r         - Stratification information (of class histogram)
#
# Notes:
#   Includes options to use the extended Ekman rule, and a kernel
#   estimator for the cumulative distribution function.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/02/2009  M. Hayward      Original code
#   19/09/2009  M. Hayward      Update and improve code efficiency
#   27/02/2010  M. Hayward      Added additional approaches
#

ekman <- function(x, strata = 3, freq = NULL, method = c("stepfun",
    "extended", "kernel"), plot = FALSE, ...){

    ###################################################################
    # Check data and calculate frequencies

    # Calculate freq if missing
    if (missing(freq)){

        # Ensure x-values are sorted
        x = sort(x)

        # Calculate cumulative frequencies of x
        cs = 1:length(x)
```

```
# Check order of freq if specified
} else {

    # Find order of x-values
    xorder = order(x)

    # Ensure x-values are sorted
    x = x[xorder]

    # Ensure frequency values are in the correct order
    freq = freq[xorder]

    # Calculate cumulative frequencies
    cs = cumsum(freq)

}

#########################################################################

#########################################################################
# Find boundaries

# Match method argument
method <- match.arg(tolower(method), c("stepfun", "extended",
    "kernel"))

# Match breaks function
fn <- switch(method,

    # Stepwise function
    "stepfun" = ekman.stepfun,

    # Extended Ekman function
    "extended" = ekman.extended,

    # Kernel-based Ekman function
    "kernel" = ekman.kernel,

    # Unknown method
    stop("unknown 'breaks' algorithm")

)

# Find boundary points (data structure of x-y coordinates)
xy = ekman.boundary(x,cs,strata=strata,fn=fn)

#########################################################################

#########################################################################
```

```
    # Set-up output

    # Construct histogram object
    r = bins(x,breaks=xy$x)

    ##########################################################################

    ##########################################################################
    # Output results

    # Output graphs for ekman function
    if (plot){

        # Plot function of csf versus boundaries
        ekman.plot(x=x,xy=xy,method=method,...)

        # Return results
        invisible(r)

    } else {

        # Return results
        r

    }

    ##########################################################################

}

##############################################################################




##############################################################################
# R Function: ekman.boundary
#
# Purpose:
#   Estimates optimal boundaries for stratified sampling using the
#   Ekman algorithm.
#
# Define variables:
#   Input:
#     x         - Vector of x-axis values (i.e. the raw values)
#     y         - Vector of y-axis values (i.e. cumulative frequency)
#     strata    - Vector of frequencies for the given data
#     method    - Boundary function
#     x0        - The first x value (default is the first value)
```

```
#     y0        - The first y value (default is the first value)
#     maxit     - Maximum number of iterations
#     tol       - Tolerance level
#   Output:
#     xy        - Coordinates of the boundaries
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   19/09/2009  M. Hayward      Original code (from Ekman function)
#   27/02/2010  M. Hayward      Added extensions
#

ekman.boundary <- function(x, y, strata, fn = ekman.stepfun,
    x0 = x[1], y0 = 0, maxit = 100, tol = 1e-7){

    ######################################################################
    # Check input values

    # Check value of n
    if (length(strata) > 1 || !is.numeric(strata) ||
        !is.finite(strata) || strata < 1)
        stop("invalid value for 'strata'")

    # Calculate y values (if not specified)
    if (missing(y))
        y <- 1:length(x)

    ######################################################################

    ######################################################################
    # Initialise values

    # Find last x and y values
    xN <- x[length(x)]
    yN <- y[length(y)]

    # Initialise the resulting boundary points
    xy <- structure(list(x = c(rep.int(x0,strata),xN),
        y = c(rep.int(y0,strata),yN)))

    ######################################################################

    ######################################################################
    # Start recursive search process (for more than one strata)

    if (strata >= 2){

        ##################################################################
```

```
# Decide on start values

# Set initial lower and upper bounds
lxy <- structure(list(x = c(rep.int(x0,strata),xN),
    y = c(rep.int(y0,strata),yN)))
uxy <- structure(list(x = c(x0,rep.int(xN,strata)),
    y = c(y0,rep.int(yN,strata))))

# Set initial lower and upper area
larea = 0
uarea = (xN - x0) * (yN - y0)

#####################################################################

#####################################################################
# Find boundaries

# Initialise iterations
iter <- 0

# Recursively search for optimal boundary points
while ((uarea - larea) > (tol^2 * uarea) & iter < maxit){

    # Calculate new area
    area <- (uarea - larea) / 2 + larea

    # Find boundaries for the given area
    for(j in 2:strata){
        val <- fn(x=x,y=y,area=area,x0=xy$x[j-1],y0=xy$y[j-1])
        xy$x[j] <- val[1]
        xy$y[j] <- val[2]
    }

    # Calculate the area for each stratum
    ab <- diff(xy$x)*diff(xy$y)

    # If calculated area is too small
    if (area < ab[strata]){
        # Area needs to be larger
        larea <- area
        # Retain boundaries
        lxy = xy

    # If calculated area is too large
    } else {
        # Area needs to be smaller
        uarea <- area
        # Retain boundaries
        uxy = xy
```

```
            }

            # Increament iterations
            iter <- iter + 1

        }

        # Calculate the area for the final upper and lower bounds
        larea = diff(lxy$x)*diff(lxy$y)
        uarea = diff(uxy$x)*diff(uxy$y)

        # Find the best set of boundary points
        if(var(larea) <= var(uarea)){
            xy = lxy
        } else {
            xy = uxy
        }

        ######################################################################

    }

    ##########################################################################

    ##########################################################################
    # Output results

    # Return results
    xy

    ##########################################################################

}

##############################################################################




##############################################################################
# R Function: ekman.stepfun
#
# Purpose:
#   To find the points closest to the given area for the Ekman
#   algorithm.
#
# Define variables:
#   Input:
#     x         - Vector of x-axis values (i.e. the raw values)
```

```
#     y         - Vector of y-axis values (i.e. cumulative frequency)
#     area      - The area objective
#     x0        - The first x value (default is zero)
#     y0        - The first y value (default is the first value)
#   Output:
#     xy        - Coordinates closest to the given area
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   19/09/2009  M. Hayward      Original code
#

ekman.stepfun <- function(x, y, area, x0 = x[1], y0 = 0){

    ########################################################################
    # Check input values

    # Check length of x and y
    if (length(x) != length(y)){
        stop("length of x and y must be the same")
    }

    # Check area is specified
    if (missing(area)) {
        stop("area must be specified")
    }

    # Find values greater than or equal to the start points
    wxy <- which(x >= x0 & y >= y0)
    xb <- x[wxy]
    yb <- y[wxy]

    ########################################################################

    ########################################################################
    # Find intervals

    # Calculate area using given start points
    ab <- (xb - x0) * (yb - y0)

    # Find the index closest to the given area
    wb <- which.min(abs(ab-area))

    ########################################################################

    ########################################################################
    # Output result
```

```
    # Calculate the resulting boundaries
    xy <- c(xb[wb],yb[wb])

    # Return results
    xy

    ########################################################################

}

##############################################################################




##############################################################################
# R Function: ekman.extended
#
# Purpose:
#   To find the coordinates corresponding to the given area using the
#   extended Ekman approach.
#
# Define variables:
#   Input:
#     x         - Vector of x-axis values (i.e. the raw values)
#     y         - Vector of y-axis values (i.e. cumulative frequency)
#     area      - The area objective
#     x0        - The first x value (default is zero)
#     y0        - The first y value (default is the first value)
#   Output:
#     xy        - Coordinates closest to the given area
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   27/02/2010  M. Hayward      Original code
#

ekman.extended <- function(x, y, area, x0 = x[1], y0 = 0){

    ########################################################################
    # Check input values

    # Check length of x and y
    if (length(x) != length(y)){
        stop("length of x and y must be the same")
    }

    # Check area is specified
```

```
if (missing(area)) {
    stop("area must be specified")
}

# Find points greater than or equal to start values
wxy <- which(x >= x0 & y >= y0)
xb <- c(x0,x[wxy])
yb <- c(y0,y[wxy])

#########################################################################

#########################################################################
# Find intervals

# Calculate area at the original points (similar to ekman.stepfun)
ab <- (xb - x0) * (yb - y0)

# Find point less than or equal to area
wb <- findInterval(area,ab)

# Calculate area at the intersection of the adjoining lines
ab <- (xb - x0) * (c(y0,yb[-length(yb)]) - y0)

# Find point less than or equal to area
wb0 <- findInterval(area,ab)

#########################################################################

#########################################################################
# Calculate the resulting boundaries

# Area results in a point on a horizontal line
if (wb == wb0){
    xy <- c(min(max(x),area/(yb[wb]-y0)+x0),yb[wb])

# Area results in a point on a vertical line
} else {
    xy <- c(xb[wb0],min(max(y),area/(xb[wb0]-x0)+y0))
}

#########################################################################

#########################################################################
# Output result

# Return results
xy

#########################################################################
```

```
}

##############################################################################




##############################################################################
# R Function: ekman.kernel
#
# Purpose:
#   To find the coordinates corresponding to the given area using the
#   Ekman approach on a kernel density function.
#
# Define variables:
#   Input:
#     x         - Vector of x-axis values (i.e. the raw values)
#     y         - Vector of y-axis values (i.e. cumulative frequency)
#     area      - The area objective
#     x0        - The first x value (default is the first value)
#     y0        - The first y value (default is the first value)
#   Output:
#     xy        - Coordinates closest to the given area
#
# Notes:
#   Includes option to construct bounded kernel estimates (default).
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   27/02/2010  M. Hayward      Original code
#

ekman.kernel <- function(x, y, area, x0 = x[1], y0 = 0,
    bounds = c(x[1], x[length(x)]), maxit = 100){

    ##########################################################################
    # Check input values

    # Check length of x and y
    if (length(x) != length(y)){
        stop("length of x and y must be the same")
    }

    # Check area is specified
    if (missing(area)) {
        stop("area must be specified")
    }
```

```
# Find frequency
freq <- round(c(y[1],diff(y)))

# Enumerate x-values
if (any(freq != 1))
    x <- rep.int(x,freq)

########################################################################

########################################################################
# Find intervals

# Initialise return values and associated area
xy <- c(x0,y0)
ab <- 0

# Set lower and upper starting values
lxy <- xy
uxy <- c(x[length(x)],y[length(y)])

# Set iterations
iter = 0

# Find area using bisection method
while (abs(ab - area) > (1e-7) & iter < maxit){

    # Find new x-value
    xy <- (lxy + uxy) / 2

    # Find new y-value
    xy[2] <- kernel.cdf(x=xy[1],data=x,bounds=bounds) *
        length(x)

    # Calculate the area
    ab <- (xy[1] - x0) * (xy[2] - y0)

    # If calculated area (ab) is too small
    if (ab < area){
        # Retain boundaries
        lxy <- xy

    # If calculated area (ab) is too large
    } else {
        # Retain boundaries
        uxy <- xy
    }

    # Increment iterations
```

```
        iter <- iter + 1

    }

    ########################################################################

    ########################################################################
    # Output result

    # Return results
    xy

    ########################################################################

}

##############################################################################




##############################################################################
# R Function: ekman.plot
#
# Purpose:
#   Plots diagnostic graphs for boundaries created by the Ekman
#   algorithm.
#
# Define variables:
#   Input:
#     x         - Raw data
#     xy        - Coordinates that define strata
#     method    - Boundary method
#     do.points - Option to plot interval points
#     main      - Main title for plots
#     xlab      - Label for x-axis
#     ylab      - Label for cumulative frequency axis
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/02/2009  M. Hayward      Original code
#   19/09/2009  M. Hayward      Separated out function
#   27/02/2010  M. Hayward      Added additional approaches
#

ekman.plot <- function(x, xy, method = "stepfun", do.points = FALSE,
    main = "", xlab = "Auxiliary Variable",
    ylab = "Cumulative distribution"){
```

```
    ######################################################################
    # Set-up breaks function

    # Calculate cumulative square root of frequencies (cs)
    cs <- seq(0,1,by=1/length(x))

    # Match method argument
    method <- match.arg(tolower(method), c("stepfun", "extended",
        "kernel"))

    ######################################################################

    ######################################################################
    # Plot function of csf versus boundaries

    # Set up plot area
    plot(xy,xlim=range(x),ylim=range(cs),main=main,xlab=xlab,
        ylab=ylab,type='n')

    # Plot Ekman rectangles
    for (i in 2:length(xy$x)){
        polygon(c(xy$x[(i-1)],xy$x[(i-1)],xy$x[i],xy$x[i]),
            c(xy$y[(i-1)],xy$y[i],xy$y[i],xy$y[(i-1)])/length(x),
            col=3,border=NA)
    }

    # Plot lines
    if (method == "kernel") {
        dr <- 0.2*diff(range(x))
        xi <- c(min(x)-dr,seq(min(x),max(x),length.out=100),max(x)+dr)
        lines(xi,kernel.cdf(x=xi,data=x,bounds=c(min(x),max(x))))
    } else {
        lines(stepfun(x,y=cs),do.points=do.points)
    }

    ######################################################################

}

######################################################################
```

B.2.3 Other Algorithms

```
######################################################################
```

```
# R Function: cfreq
#
# Purpose:
#   Constructs stratum boundaries using equal intervals on the
#   cumulative frequency scale.
#
# Define variables:
#   Input:
#     x         - Raw data
#     strata    - Number of strata
#   Output:
#     st        - Stratification information (of class histogram)
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   12/12/2009  M. Hayward      Original code
#

cfreq <- function(x, strata = 6){

    ###########################################################################
    # Set-up variables

    # Check number of strata
    if (length(strata) > 1 || !is.numeric(strata) ||
        !is.finite(strata) || strata < 1)
        stop("invalid number of 'strata'")

    # Obtain the unevaluated expression for x and turn into a string
    xname <- paste(deparse(substitute(x), 500), collapse = "\n")

    ###########################################################################

    ###########################################################################
    # Construct strata

    # Ensure x-values are sorted
    x = sort(x)

    # Construct frequency breaks
    yb <- seq(0,length(x),length.out=strata+1)

    # Find break indicies
    wb <- c(1,findInterval(yb[-1],1:length(x)))

    # Calculate summary information
    st <- bins(x,breaks=x[wb])
```

```
    # Return results
    st

    ##########################################################################

}

##############################################################################




##############################################################################
# R Function: eqint
#
# Purpose:
#   Constructs stratum boundaries using equal intervals along the
#   range of the data.
#
# Define variables:
#   Input:
#     x         - Raw data
#     strata    - Number of strata
#   Output:
#     st        - Stratification information (of class histogram)
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   12/12/2009  M. Hayward      Original code
#

eqint <- function(x, strata = 6){

    ##########################################################################
    # Set-up variables

    # Check number of strata
    if (length(strata) > 1 || !is.numeric(strata) ||
        !is.finite(strata) || strata < 1)
        stop("invalid number of 'strata'")

    # Obtain the unevaluated expression for x and turn into a string
    xname <- paste(deparse(substitute(x), 500), collapse = "\n")

    ##########################################################################

    ##########################################################################
    # Construct strata
```

```
    # Construct breaks
    xb <- seq(min(x),max(x),length.out=strata+1)

    # Calculate summary information
    st <- bins(x,breaks=xb)

    # Return results
    st

    ##########################################################################

}

##############################################################################




##############################################################################
# R Function: geometric
#
# Purpose:
#   Constructs stratum boundaries using the geometric algorithm.
#
# Define variables:
#   Input:
#     x         - Raw data
#     strata    - Number of strata
#   Output:
#     st        - Stratification information (of class histogram)
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   20/06/2009  M. Hayward      Original code
#

geo <- function(x, strata = 6){

    ##########################################################################
    # Set-up variables

    # Check number of strata
    if (length(strata) > 1 || !is.numeric(strata) ||
        !is.finite(strata) || strata < 1)
        stop("invalid number of 'strata'")

    # Obtain the unevaluated expression for x and turn into a string
```

```
    xname <- paste(deparse(substitute(x), 500), collapse = "\n")

    ##########################################################################

    ##########################################################################
    # Construct breaks

    # Find first and last values
    a <- min(x)
    xN <- max(x)

    # Calculate the common ratio
    r <- (xN/a)^(1/strata)

    # Calculate geometric series
    xb <- c(a, a*r^(1:(strata-1)), xN)

    ##########################################################################

    ##########################################################################
    # Construct strata

    # Calculate summary information
    st <- bins(x,breaks=xb)

    # Return results
    st

    ##########################################################################

}

##############################################################################
```

B.3 Sample Algorithms

```
##############################################################################
# R Function: allocate
#
# Purpose:
#   Finds the number of values to sample from each strata.
#
# Define variables:
#   Input:
```

```
#     n         - Sample size
#     Nh        - Stratum size
#     Sh        - Stratum standard deviation
#     Ch        - Stratum sample cost
#     min       - Minimum stratum sample
#     maxit     - Maximum number of iterations
#   Output:
#     nh        - Number to sample from each stratum
#
# Notes:
#   Uses optimal allocation to allocate values among strata. Neyman
#   allocation is achieved through setting Ch to 1, and proportional
#   allocation through setting Sh and Ch to 1.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   23/02/2009  M. Hayward      Original code
#   13/07/2009  M. Hayward      Combined allocate code
#

allocate <- function(n, Nh, Sh = 1, Ch = 1, min = 2, maxit = 100){

    ######################################################################
    # Check input values

    # Ensure n is a whole number
    n <- round(n)

    # Check if sufficient sample size
    if(n > sum(Nh))
        stop("'n' greater than population size 'Nh'")

    # Check if minimum greater than or equal to zero
    if(min < 0)
        stop("'min' must be greater than zero")

    # Check if minimum sample is greater than stratum size
    if(any(Nh < min))
        stop("'min' sample size greater than stratum size")

    # Check if sufficient sample size
    if(n < (length(Nh) * min))
        stop("'min' sample size greater than 'n'")

    ######################################################################

    ######################################################################
    # Calculate number to sample
```

```
# Calculate the adjusted weighting for each stratum
Ah <- Nh * Sh / sqrt(Ch)

# Calculate adjusted overall sampling fraction
Af <- n / sum(Ah)

# Calculate number to sample from each stratum (round to nearest
# value)
nh <- round(Ah * Af)

# If values less than the minimum, set to the minimum
nh[nh < min] <- min

# If values greater than stratum size, set to the stratum size
nh[nh > Nh] <- Nh[nh > Nh]

######################################################################

######################################################################
# Sample size corrections

# Initialise value for interations of while loop
it <- 0

# Recursively adjust nh values
while(abs(n - sum(nh)) > 1e-07){

    # If sum of sample from strata is less than sample size
    if(n > sum(nh)){

        # Add value to nh
        nh1 <- nh + 1

        # Restrict adjustment to values less than Nh
        val <- nh1 <= Nh

        # Find the smallest increase in frequency
        ind <- which.min(nh1[val]/Ah[val] - Af)

    # If sum of sample from strata is greater than sample size
    } else {

        # Subtract value from nh
        nh1 <- nh - 1

        # Restrict adjustment to values greater than min
        val <- nh1 >= min
```

```
            # Find the smallest decrease in frequency
            ind <- which.max(nh1[val]/Ah[val] - Af)

        }

        # Adjust value (must used *[val][ind])
        nh[val][ind] <- nh1[val][ind]

        # Check and advance loop interations
        if((it <- it + 1) >= maxit)
            warning("Maximum number of iterations reached")

    }

    ##########################################################################

    ##########################################################################
    # Output results

    # Output sample sizes
    nh

    ##########################################################################

}

##############################################################################




##############################################################################
# R Function: sample.strata
#
# Purpose:
#   Samples values from strata
#
# Define variables:
#   Input:
#     x         - Dataset (to sample)
#     h         - Stratum corresponding to 'x'
#     Nh        - Stratum population sizes
#     n         - Sample size
#     method    - Stratum sample method
#     allocate  - Allocation of sample between strata
#     ...       - Further arguments to allocate
#   Output:
#     r         - Output
#
```

```
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/02/2009  M. Hayward      Original code
#   14/04/2009  M. Hayward      Update sample method
#

sample.strata <- function(x, h, Nh, n, method = "srswor",
    allocate = "proportional", ...){

    ######################################################################
    # Initial set up and checking

    # Check for stratum information
    if (missing(h) & missing(Nh))
        stop("'h' or 'Nh' must be specified")

    # Check for stratum indicator
    if (missing(h))
        h <- rep.int(1:length(Nh),Nh)

    # Check for stratum size
    if (missing(Nh))
        Nh <- tabulate(h)

    # Check population
    if (missing(x)) {
        x <- 1:length(h)
    } else {
        # Test if population is numeric
        if (!is.numeric(x))
            stop("'x' must be numeric")
    }

    # Check population size
    if (length(x) != sum(Nh) || length(x) != length(h))
        stop("'x' does not match stratum size")

    # Check sample size
    if (length(n) > 1 || !is.numeric(n) || !is.finite(n) || n < 1 ||
        n > length(x))
        stop("invalid sample size 'n'")

    ######################################################################

    ######################################################################
    # Calculate sample size

    # Find allocation algorithm
```

```
    allocate <- match.arg(tolower(allocate),c("proportional",
        "optimal","neyman"))

    # Calculate allocation
    nh <- switch(allocate,
        "proportional"  = allocate(n=n,Nh=Nh,Sh=1,Ch=1, ...),
        "neyman"        = allocate(n=n,Nh=Nh,Sh=1, ...),
        "optimal"       = allocate(n=n,Nh=Nh, ...),
        stop("unknown allocation algorithm")
    )

    ######################################################################

    ######################################################################
    # Sample strata and return results

    # Construct dataset
    data <- cbind.data.frame(x,h)

    # Sample strata
    s <- strata(data,"h",size=nh,method=method)

    # Construct output
    r <- structure(list(sample=x[s$ID_unit],index=s$ID_unit,
        stratum=s$Stratum,prob=s$Prob,nh=nh,Nh=Nh,allocate=allocate))

    # Output results
    r

    ######################################################################

}

##########################################################################




##########################################################################
# R Function: summary.strata
#
# Purpose:
#   To summarise results from stratified random sampling.
#
# Define variables:
#   Input:
#     x         - An object resulting from sample.strata
#     fpc       - Option for finite population correction
#   Output:
```

```
#     r         - Output
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/02/2009  M. Hayward      Original code
#   14/04/2009  M. Hayward      Update sample method
#

summary.strata <- function(x, fpc = TRUE){

    ######################################################################
    # Initial set up and checking

    # Check stratum sample and population sizes
    if(length(x$nh) != length(x$Nh) || any(x$nh > x$Nh))
        stop("invalid values of 'nh' or 'Nh'")

    # Calculate sample size and population size
    n <- sum(x$nh)
    N <- sum(x$Nh)

    # Calculate finite population correction
    # Sets stratum variance to zero for take-all stratum
    fpc.val <- if (fpc){1 - x$nh / x$Nh} else {1}

    ######################################################################

    ######################################################################
    # Calculate stratum statistics

    # Calculate mean for each stratum
    meanh <- tapply(x$sample,x$stratum,mean)

    # Calculate stratum variance
    varh <- tapply(x$sample,x$stratum,var) / x$nh * fpc.val

    ######################################################################

    ######################################################################
    # Calculate overall statistics

    # Calculate sample mean and variance
    meanst <- sum(meanh * x$Nh / N)
    varst <- sum(varh * (x$Nh / N)^2)

    # Calculate variance of a simple random sample
    int1 <- (N - n) / (n * (N - 1))
    int2 <- sum(tapply(x$sample^2,x$stratum,mean) * x$Nh) / N
```

```
    vran <- int1 * (int2 - meanst^2 + varst)

    # Calculate design effect
    deff <- varst / vran

    ########################################################################

    ########################################################################
    # Summarise and output results

    # Construct output
    r <- structure(list(mean=meanst, var=varst, strata.mean=meanh,
        strata.var=varh, srs.var=vran, design.effect=deff))

    # Output results
    r

    ########################################################################

}

##############################################################################
```

B.4 Supplementary Functions

```
##############################################################################
# R Function: findBoundary
#
# Purpose:
#   Finds closest indices of tb (x) in cs (vec), where cs (vec) must
#   be sorted (non-decreasingly).
#
# Define variables:
#   Input:
#     tb        - Vector of breaks
#     cs        - Vector of counts
#     no.dups   - Adjust for duplicate values equal to breaks
#   Output:
#     wb        - Output
#
# Notes:
#   Based on findInterval, but corrects for closest values and exact
#   matches. In particular returns the first value for an exact match,
#   whereas findInterval will return the last value (if there are
```

```
#   duplicate exact matches).
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   13/09/2009  M. Hayward      Original code
#   28/11/2009  M. Hayward      Speed improvements
#

findBoundary <- function(tb,cs,no.dups=TRUE){

    # Find possible values (cs[wb[j]] <= tb[j] < cs[wb[j]+1])
    # (setting all.inside ensures indices mapped to {1,...,N-1}
    wb <- findInterval(tb,cs,all.inside=TRUE)

    # Find next indicies (requires last index to be mapped to N-1)
    wb1 <- wb+1

    # Ensure minimum distance to intervals (left bias for equal
    # distance)
    wr <- abs(cs[wb1]-tb) < abs(cs[wb]-tb)

    # Update values with minimum distance points
    wb[wr] <- wb1[wr]

    # Correct for duplicate values on theoretical boundaries
    while (no.dups){

        # Find empirical boundaries that match theoretical boundaries
        # Must match by position and value (hence requires "==")
        if(!any(m <- cs[wb]==tb)) break

        # Find any duplicate values
        if(length(d <- which(duplicated(cs))) < 1) break

        # Find any boundary matches in the set of duplicate values
        if(!any(j <- wb[m]%in%d)) break

        # Get unique values (required to adjust final values)
        u <- (1:length(cs))[-d]

        # Find new boundary values and exit from loop
        # Must adjust for unique values (as cs subset by u)
        # Index must be specifed [m][j], and not [m[j]]
        wb[m][j] <- u[findBoundary(tb[m][j],cs[u],no.dups=FALSE)]
        break

    }
```

```
    # Return result
    wb

}

################################################################################




################################################################################
# R Function: kernel.cdf
#
# Purpose:
#   To find the cumulative distribution function values corresponding
#   to the given values using a kernel density estimator.
#
# Define variables:
#   Input:
#     x         - Values for the evaluation of the CDF kernel
#     data      - Vector of data values
#     bounds    - Bounds on CDF values
#     h         - Smoothing parameter
#   Output:
#     fhat      - CDF kernel values (corresponding to x)
#
# Notes:
#   Re-normalises bounded kernel density for each data point.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   27/02/2010  M. Hayward      Original code
#

kernel.cdf <- function(x,data,bounds=c(-Inf,Inf),h){

    # Length of data
    n <- length(data)

    # Set up x values (this will be irrelavant)
    first <- rep.int(0,length(x[x < bounds[1]]))
    last <- rep.int(1,length(x[x > bounds[2]]))
    x <- x[x >= bounds[1] & x <= bounds[2]]

    # Initialise resulting vector
    fhat = numeric(length(x))

    # Obtain smoothing parameter
```

```
    if (missing(h)) {
        if (sd(data) > 0){
            h = min(1.06 * sd(data) * n^(-0.2),
                0.786 * IQR(data) * n^(-0.2))
        } else {
            h = 1.06 * n^(-0.2)
        }
    }

    # Loop through data points
    for (i in 1:length(x)){

        # Evaluate kernel function at x
        f = pnorm(x[i],mean=data,sd=h)-pnorm(bounds[1],mean=data,sd=h)

        # Adjust kernel estimate
        bhat = pnorm(bounds[2],mean=data,sd=h)-pnorm(bounds[1],
            mean=data,sd=h)
        f[bhat > 0] = (f / bhat)[bhat > 0]

        # Add to result
        fhat[i] = sum(f)/n
    }

    # Format result
    fhat = c(first,fhat,last)
    fhat

}

##############################################################################




##############################################################################
# R Function: bins
#
# Purpose:
#   Allocates values to number or vector of bins.
#
# Define variables:
#   Input:
#     x         - Dataset
#     breaks    - Number or vector of breaks
#     inc.low   - Include lowest point of breaks
#     right     - Breaks are right inclusive
#     tol       - Tolerance level (for equidist & fuzz)
#     fuzz      - Fuzzy breaks
```

```
#   Output:
#     r         - Output
#
# Notes:
#   Largely based on the hist function, with changes to increase speed
#   and efficiency. Includes direct control of tolerance (tol) and
#   fuzzybreaks (fuzz).
#
#   Tolerance is set to hist values, and fuzzybreaks are set to zero
#   by default. Produces identical results to hist when fuzz is not a
#   numeric value (e.g. fuzz = TRUE).
#
#   Does not produce plots or support break algorithms. Both can still
#   be used through additional statements before or after calling the
#   bins function.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   08/02/2009  M. Hayward      Original code
#   08/03/2009  M. Hayward      Made similar to hist
#   31/05/2009  M. Hayward      Final documentation
#

bins <- function(x, breaks = 10, include.lowest = TRUE, right = TRUE,
        tol = 1e-7, fuzz = 0){

    ###########################################################################
    # Initial set up and checking

    # Test if population is numeric
    if (!is.numeric(x))
        stop("'x' must be numeric")

    # Obtain the unevaluated expression for x and turn into a string
    xname <- paste(deparse(substitute(x), 500), collapse = "\n")

    # Determine the number of finite values in x (n)
    n <- length(x <- x[is.finite(x)])

    ###########################################################################

    ###########################################################################
    # Set up vector of breaks

    # Check if vector of breaks specified (length > 1)
    use.br <- (nB <- length(breaks)) > 1

    # If vector of breaks then sort (use.br == TRUE)
```

```
if (use.br)
    breaks <- sort(breaks)

# If breaks are not a vector then create vector
else {

    # Check if option to include lowest is incorrectly specified
    if (!include.lowest) {
        include.lowest <- TRUE
        warning(paste("'include.lowest' ignored as 'breaks' is",
            "not a vector"))
    }

    # Check breaks are numeric, fininte, and greater than zero
    if (!is.numeric(breaks) || !is.finite(breaks) || breaks < 1)
        stop("invalid number of 'breaks'")

    # Create vector of breaks
    breaks <- seq(min(x),max(x),length.out=breaks+1)

    # Determine the number of breaks
    nB <- length(breaks)

}

# Find interval between breaks
h <- diff(breaks)

# Determine if equal distance between breaks
equidist <- !use.br || diff(range(h)) < tol * mean(h)

# Check if created breaks are strictly increasing
if (!use.br && any(h <= 0))
    stop("'breaks' are not strictly increasing")

#############################################################################

#############################################################################
# Count frequency for bins

# Calculate fuzzy breaks
diddle <- tol * if(!is.numeric(fuzz)) stats::median(diff(breaks))
    else fuzz
fuzz <- if (right) c(if (include.lowest) -diddle else diddle,
        rep.int(diddle, length(breaks) - 1))
    else c(rep.int(-diddle, length(breaks) - 1),
        if (include.lowest) diddle else -diddle)
fuzzybreaks <- breaks + fuzz
```

```
    # Set storage parameters for bincounts function
    storage.mode(x) <- "double"
    storage.mode(fuzzybreaks) <- "double"

    # Count frequency of values in bins
    counts <- .C("bincount", x, as.integer(n), fuzzybreaks,
        as.integer(nB), counts = integer(nB - 1), right =
        as.logical(right), include = as.logical(include.lowest), naok
        = FALSE, NAOK = FALSE, DUP = FALSE, PACKAGE = "base")$counts

    # Check bin counts are greater than zero
    if (any(counts < 0))
        stop(paste("negative 'counts'. Internal Error in C-code for",
            "\"bincount\""))

    # Check sum of counts equal length of x
    if (sum(counts) < n)
        stop(paste("some 'x' not counted; maybe 'breaks' do not span",
        "range of 'x'"))

    ##########################################################################

    ##########################################################################
    # Return information

    # Calculate densities and midpoints of breaks
    dens <- counts/(n * h)
    mids <- 0.5 * (breaks[-1] + breaks[-nB])

    # Set up return value
    r <- structure(list(breaks = breaks, counts = counts,
        intensities = dens, density = dens, mids = mids, xname =
        xname, equidist = equidist), class = "histogram")

    # Return result
    r

    ##########################################################################

}

##############################################################################




##############################################################################
# R Function: as.hist
#
```

```
# Purpose:
#   Constructs an object of class histogram.
#
# Define variables:
#   Input:
#     breaks    - Vector of breaks
#     counts    - Vector of counts
#     xname     - Variable name
#   Output:
#     r         - Output
#
# Notes:
#   Calculates densities and midpoints for a histogram object using a
#   vector of breaks, and a vector counts of values within each
#   consective set of break points.
#
#   The length of vecor of counts should equal length(breaks) - 1, and
#   the value of xname will default to the name of the breaks variable
#   if it is not specified.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   31/05/2009  M. Hayward      Original code
#

as.hist <- function(breaks, counts, xname){

    ########################################################################
    # Initial set-up and checking

    # Check if breaks or counts are missing
    if (missing(breaks) || missing(counts)) {
        stop("'breaks' and 'counts' must be specifed")
    }

    # Check length of breaks and counts
    if (length(breaks) != length(counts) + 1) {
        stop("invalid length of 'breaks' or 'counts'")
    }

    # Obtain if xname is missing
    if (missing(xname)) {
        xname <- paste(deparse(substitute(breaks),500),collapse="\n")
    }

    ########################################################################

    ########################################################################
```

```
    # Construct histogram object

    # Find interval between breaks
    h <- diff(breaks)

    # Determine if equal distance between breaks
    equidist <- diff(range(h)) < 1e-07 * mean(h)

    # Calculate densities and midpoints of breaks
    dens <- counts / (sum(counts) * h)
    mids <- 0.5 * (breaks[-1] + breaks[-length(breaks)])

    # Set up return value
    r <- structure(list(breaks = breaks, counts = counts,
        intensities = dens, density = dens, mids = mids, xname =
        xname, equidist = equidist), class = "histogram")

    # Return result
    r

    ##########################################################################

}

##############################################################################
```

B.5 Population Simulations

```
##############################################################################
# R Function: mvrlnorm
#
# Purpose:
#   Produces a multivariate log-normal distribution for a specified
#   correlation matrix.
#
# Define variables:
#   Input:
#     n         - Number of observations
#     mu        - Log-mean (default = 0)
#     sigma     - Log standard deviation (default = 1)
#     cormat    - Correlation matrix (not on the log scale)
#     tol       - Tolerance value (relative to largest variance)
#   Output:
#     y         - Resulting population
```

```
#
# Notes:
#   Correlation matrix is a square matrix with the number of rows and
#   columns equal to the number of elements in mu/sigma.
#
# Record of revisions:
#   Date        Programmer      Description of change
#   ====        ==========      =====================
#   06/06/2009  M. Hayward      Original code.
#

mvrlnorm <- function(n = 1, mu = 0, sigma = 1, cormat, tol = 1e-09){

    ###################################################################
    # Check input values and set default values

    # If n is a vector, set n equal to the length of n
    if(length(n) != 1 | !is.numeric(n))
        n <- length(n)

    # If n is less than one or is NA, return an empty numeric value
    if(n < 1 | is.na(n))
        return(numeric())

    # Check mu and sigma have the same number of values.
    if(length(mu) != length(sigma))
        stop("mu and sigma do not have equal number of values")

    # Check correlation matrix and create one if missing.
    if(missing(cormat)){
        cormat <- matrix(0,nrow=length(mu),ncol=length(mu))
        warning("Correlation matrix is missing")
        diag(cormat) <- 1
    } else {

        # Check number of rows and colums.
        if(nrow(cormat) != length(mu))
            stop("correlation matrix dimension mismatch")

        # Check number of rows and colums.
        if(nrow(cormat) != ncol(cormat))
            stop("correlation matrix dimension mismatch")

        # Check symmetry of matrix.
        if(all(abs(cormat - t(cormat)) > tol))
            stop("correlation matrix is not symmetric")

        # Check correlations (valeus between -1 and 1).
        if(all(abs(cormat) - 1 > tol))
```

```
            stop("correlation matrix values out of range")

        # Check diagonal values (diagonal values equal 1).
        if(all(abs(diag(cormat) - 1) > tol))
            stop("correlation matrix main diagonal not equal to one")

    }

    ######################################################################

    ######################################################################
    # Calculate values

    # Calculate the covariance structure
    sigma_down = matrix(rep(sigma,length(sigma)),nrow=length(sigma),
        byrow=TRUE)
    sigma_acrs = matrix(rep(sigma,length(sigma)),nrow=length(sigma),
        byrow=FALSE)

    covv = log(cormat * sqrt(exp(sigma_down^2)-1) *
        sqrt(exp(sigma_acrs^2)-1) + 1)

    # Simulate values
    y = exp(mvrnorm(n,mu,covv))

    # Return result
    y

    ######################################################################

}

######################################################################
```

References

Baillargeon, S., Rivest, L.-P. & Ferland, M. (2007), Stratification en enquêtes entreprises: Une revue et quelques avancées, *in* 'Proceedings of the Survey Methods Section, 2007 SSC Annual Meeting'.

Chambers, R. L. (1996), 'Robust case-weighting for multipurpose establishment surveys', *Journal of Official Statistics* **12**(1), 3–32.

Cochran, W. G. (1961), 'Comparison of methods for determining stratum boundaries', *Bulletin of the International Statistical Institute* **38**, 345–358.

Cochran, W. G. (1977), *Sampling techniques*, 3rd edn, John Wiley & Sons, New York, USA.

Dalenius, T. (1950), 'The problem of optimum stratification', *Skandinavisk Aktuarietidskrift* **33**, 203–213.

Dalenius, T. (1957), *Sampling in Sweden: contributions to the methods and theories of sample survey practice*, Almqvist & Wiksell, Stockholm, Sweden.

Dalenius, T. & Hodges, J. L. (1957), 'The choice of stratification points', *Skandinavisk Aktuarietidskrift* **40**, 198–203.

Detlefsen, R. E. & Veum, C. S. (1991), Design issues for the retail trade sample surveys of the U.S. Bureau of the Census, *in* 'Proceedings of the Survey Research Methods Section', American Statistical Association, pp. 214–219.

Ekman, G. (1959*a*), 'Approximate expressions for the conditional mean and variance over small intervals of a continuous distribution', *The Annals of Mathematical Statistics* **30**(4), 1131–1134.

Ekman, G. (1959*b*), 'An approximation useful in univariate stratification', *The Annals of Mathematical Statistics* **30**(1), 219–229.

Ekman, G. (1963), 'On the sum $\Sigma^n P_h^i \sigma_h^j$', *Review of the International Statistical Institute* **31**(1), 67–80.

Ekman, G. (1969), 'A graphical solution for the determination of an optimal stratification or grouping, with an example concerning a problem of maximizing sales revenues', *Review of the International Statistical Institute* **37**(2), 186–194.

Fritsch, F. N. & Carlson, R. E. (1980), 'Monotone piecewise cubic interpolation', *SIAM Journal on Numerical Analysis* **17**(2), 238–246.

Gunning, P. & Horgan, J. M. (2004), 'A new algorithm for the construction of stratum boundaries in skewed populations', *Survey Methodology* **30**(2), 159–166.

Gunning, P., Horgan, J. M. & Keogh, G. (2008), 'A new algorithm for the construction of stratum boundaries in skewed populations', *Journal of Official Statistics* **24**(2), 213–228.

Hedlin, D. (2000), 'A procedure for stratification by an extended Ekman rule', *Journal of Official Statistics* **16**(1), 15–29.

Hedlin, D. (2003), Minimum variance stratification of a finite population, SSRC Methodology Working paper M03/07, Social Statistics Research Centre, University of Southampton.

Hess, I., Sethi, V. K. & Balakrishnan, T. R. (1966), 'Stratification: a practical investigation', *Journal of the American Statistical Association* **61**(313), 74–90.

Hidiroglou, M. A. (1986), 'The construction of a self-representing stratum of large units in survey design', *The American Statistician* **40**(1), 27–31.

Horgan, J. M. (2003), 'A list-sequential sampling scheme with applications in financial auditing', *IMA Journal of Management Mathematics* **14**(1), 31–48.

Horgan, J. M. (2006), 'Stratification of skewed populations: a review', *International Statistical Review* **74**(1), 67–76.

Johnson, N. L. & Kotz, S. (1972), *Distributions in Statistics: Continuous Multivariate Distributions*, John Wiley & Sons, New York, USA.

Karlberg, F. (2000), 'Survey estimation for highly skewed populations in the presence of zeroes', *Journal of Official Statistics* **16**(3), 229–241.

Kish, L. (1965), *Survey Sampling*, John Wiley & Sons, New York, USA.

Kish, L. (1976), 'Optima and proxima in linear sample designs', *Journal of the Royal Statistical Society, Series A (General)* **139**(1), 80–95.

Kozak, M. (2004), 'Optimal stratification using random search method in agricultural surveys', *Statistics in Transition* **6**(5), 797–806.

Kozak, M. (2006), 'Optimal number of strata in agricultural surveys', *Statistics in Transition* **7**(5), 971–979.

Lavallée, P. & Hidiroglou, M. (1988), 'On the stratification of skewed populations', *Survey Methodology* **14**, 33–43.

Lednicki, B. & Wieczorkowski, R. (2003), 'Optimal stratification and sample allocation between subpopulations and strata', *Statistics in Transition* **6**(2), 287–305.

Lohr, S. L. (1999), *Sampling: design and analysis*, Duxbury Press, California, USA.

Martinez, W. L. & Martinez, A. R. (2002), *Computational Statistics Handbook with Matlab*, Chapman & Hall/CRC, Florida, USA.

Neyman, J. (1934), 'On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection', *Journal of the Royal Statistical Society* **97**(4), 558–625.

Norland, R. E. (1983), An efficient algorithm for determining strata boundaries for discrete populations using Ekman's method, *in* 'Proceedings of the Statistical Computing Section, American Statistical Association', pp. 174–176.

Rao, J. N. K. (1962), 'On the estimation of the relative efficiency of sampling

procedures', *Annals of the Institute of Statistical Mathematics* **14**(1), 143–150.

Rivest, L.-P. (2002), 'A generalization of the Lavallée and Hidiroglou algorithm for stratification in business surveys', *Survey Methodology* **28**(2), 191–198.

Särndal, C.-E., Swensson, B. & Wretman, J. (1992), *Model Assisted Survey Sampling*, Springer-Verlag, New York, USA.

Satterthwaite, F. E. (1946), 'An approximate distribution of estimates of variance components', *Biometrics Bulletin* **2**(6), 110–114.

Sethi, V. K. (1963), 'A note on optimum stratification of populations for estimating the population means', *Australian Journal of Statistics* **5**(1), 20–33.

Sigman, R. S. & Monsour, N. J. (1995), Selecting samples from list frames of businesses, *in* B. G. Cox, D. A. Binder, B. N. Chinnappa, A. Christianson, M. J. Colledge & P. S. Kott, eds, 'Business Survey Methods', John Wiley & Sons, New York, USA, pp. 133–152.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, London, UK.

Stuart, A. (1954), 'A simple presentation of optimum sampling results', *Journal of the Royal Statistical Society, Series B (Methodological)* **16**(2), 239–241.

Thompson, M. E. (1997), *Theory of Sample Surveys*, Chapman & Hall, London, UK.

Thompson, S. K. (1992), *Sampling*, John Wiley & Sons, New York, USA.

Wand, M. P. & Jones, M. C. (1995), *Kernel Smoothing*, Chapman & Hall, London, UK.

Wilson, N., Russell, D. & Wilson, B. (1993), *Size and shape of New Zealanders: New Zealand norms for anthropometric data*, Life in New Zealand Activity and Health Research Unit, University of Otago, Dunedin, New Zealand.